DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

MANAGING COMPUTER SYSTEMS DEVELOPMENT:
UNDERSTANDING THE HUMAN AND
TECHNOLOGICAL IMPERATIVES

by

Gregory S. Curtis

June 1985

Thesis Advisor: N.R. Lyons

Approved for public release; distribution is unlimited

T223072

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* Managing Computer Systems Development: Understanding the Human and Technological Imperatives		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis June 1985 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR*(s)* Gregory S. Curtis		8. CONTRACT OR GRANT NUMBER*(s)*
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943		12. REPORT DATE June 1985 13. NUMBER OF PAGES 153
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution is unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Computer Information System, Systems Development Life Cycle, Structured Analysis, Business Systems Planning, Fourth Generation Languages, Database Management System, Data Dictionary System

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This paper examines the human and technological issues that are often encountered during the development of modern computer information systems. People and technical constraints, including suggestions for minimizing negative consequences, are illustrated throughout the development life cycle. Special emphasis is placed on strategic planning, end user involvement in the requirements definition phase, and user-oriented software. (Continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

ABSTRACT (Continued)

The research consists of a review of current literature concerning techniques, methods and methodologies that are the basis for managing computer information system development. It is a collection of bits and pieces of wisdom by experts from all disciplines within the computer and managing fields. These techniques can be tailored to various scale projects having myriad objectives. The theory and practice of management methods included in this paper can be applied universally to computer projects. However, the study is directed at all U.S. Navy managers who are, or will be, involved in the transition to modern computer information systems.

Approved for public release; distribution is unlimited.

Managing Computer Systems Development: Understanding the Human and Technological Imperatives

by

Gregory S. Curtis
Lieutenant, United States Navy
B.S., United States Naval Academy, 1979

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
June 1985

ABSTRACT

This paper examines the human and technological issues that are often encountered during the development of modern computer information systems. People and technical constraints, including suggestions for minimizing negative consequences, are illustrated throughout the development life cycle. Special emphasis is placed on strategic planning, end user involvement in the requirements definition phase, and user-oriented software. The research consists of a review of current literature concerning techniques, methods and methodologies that are the basis for managing computer information system development. It is a collection of bits and pieces of wisdom by experts from all disciplines within the computer and management fields. These techniques can be tailored to various scale projects having myriad objectives. The theory and practice of management methods included in this paper can be applied universally to computer projects. However, the study is directed at all U.S. Navy managers who are, or will be, involved in the transition to modern computer information systems.

TABLE OF CONTENTS

I. INTRODUCTION 11

II. INFORMATION SYSTEMS AND GOVERNMENT REGULATIONS . . 16

A. INTRODUCTION 16

B. INFORMATION - A VITAL RESOURCE 17

C. THREE LEVELS OF INFORMATION - THREE INFORMATION SYSTEMS 18

1. Electronic Data Processing (EDP) 18

2. Management Information Systems (MIS) . . . 19

3. Decision Support Systems (DSS) 20

D. COMPUTER GROWTH BECOMES A MANAGEMENT PROBLEM 22

E. INFORMATION RESOURCE MANAGEMENT (IRM) 23

F. REGULATIONS SLOW GOVERNMENT COMPUTER DEVELOPMENT 24

1. The Brooks Act 24

2. Paperwork Reduction Act of 1980 26

3. The Federal Information Resource Management Regulation 27

G. SUMMARY 29

III. STRATEGIC INFORMATION SYSTEMS PLANNING 31

A. INTRODUCTION 31

B. PLANNING FOR CHANGE 31

1. The Systems Plan 32

2. The Hardware Plan 35

3. The Software Plan 36

4. The Staffing Plan 38

5. The Control Plan 39

C. AVOIDING FAILURE 42
D. SUMMARY 45

IV. ANALYZING THE PRESENT-PROJECTING THE FUTURE . . . 46
A. INTRODUCTION 46
B. HISTORY OF BUSINESS SYSTEMS PLANNING 47
C. BSP OBJECTIVES 49
D. KEY ACTIVITIES IN THE BSP METHODOLOGY 51
1. Gaining the Commitment 51
2. Preparing for the Study 52
3. Starting the Study 52
4. Defining Business Processes 53
5. Defining Business Data 53
6. Defining the Information Architecture . . 54
7. Analyzing Current Information Support . . 54
8. Interviewing Executives 55
9. Defining Findings and Conclusions 55
10. Determining the Architecture Priorities 55
11. Reviewing Information Resource Management (IRM) 56
12. Developing Recommendations 57
13. Reporting Results 57
E. APPLYING BSP AT FORT ORD 57
1. Background 57
2. The Need for Change 58
3. The Study 60
F. PLANNING CHANGE USING SYSTEMS ANALYSIS 62
G. THE SYSTEMS ANALYSIS APPROACH 63
1. Initial Investigation 64
2. Feasibility Study 67
3. Evaluate and Decide 70
H. BSP VS. SYSTEMS ANALYSIS 71
1. BSP Strengths 71

2. BSP Weaknesses 72
3. Systems Analysis Strengths 72
4. Systems Analysis Weaknesses 73
I. SUMMARY 74

V. SYSTEMS DEVELOPMENT METHODS AND PROJECT MANAGEMENT 75
A. INTRODUCTION 75
B. THE SYSTEMS DEVELOPMENT LIFE CYCLE 75
1. Investigation Phase 76
2. Analysis and General Design Phase 77
3. Detailed Design and Implementation Phase 77
4. Installation Phase 77
5. Review Phase 78
C. VARIATIONS TO SDLC 78
1. The Serial Approach 79
2. The Overlapping Approach 79
3. The Functional Approach 79
D. SDLC LIMITATIONS 81
E. HEURISTIC SYSTEMS DEVELOPMENT AND PROTOTYPING 83
1. Heuristic Systems Development 83
2. Prototyping 85
F. BENEFITS OF HEURISTIC AND PROTOTYPING APPROACHES 86
G. PROJECT MANAGEMENT 88
1. High Structure-Low Technology 91
2. High Structure-High Technology 92
3. Low Structure-Low Technology 93
4. Low Structure-High Technology 94
5. Project Management Software Tools 95
H. SUMMARY 97

VI. USER-ORIENTED DEVELOPMENT TOOLS 99
A. INTRODUCTION 99
B. A MANAGEMENT DILEMMA 99
1. No Standard Definition for 4GL 100
2. General Characteristics of 4GL 102
3. Fourth Generation Languages Are User Constrained 104
4. Purchasing a Fourth Generation Language Package 105
5. Future Fourth Generation Language Trends 106
C. DATABASE MANAGEMENT SOFTWARE 107
D. DATABASE TECHNOLOGY 108
1. Database Concepts 108
2. DBMS: Advantages and Disadvantages . . . 109
3. Determining a Need for a DBMS 112
4. Problems with Database Methods 116
E. DATA DICTIONARY SYSTEMS (DDS) 118
1. Data Dictionary System Features 118
2. Selecting a Data Dictionary System . . . 119
F. SUMMARY 119

VII. SUMMARY 121

APPENDIX A: QUESTIONS THAT KEY MANAGERS SHOULD ASK ABOUT DP 123

APPENDIX B: A SYSTEMS ANALYSIS CHECKLIST 130

APPENDIX C: REPRESENTATIVE SUPPLIERS OF DBMS AND 4TH GENERATION LANGUAGES 141

LIST OF REFERENCES 150

INITIAL DISTRIBUTION LIST 153

LIST OF TABLES

1. Development Method Variations 87
2. Project Management Tools 89

LIST OF FIGURES

3.1 Staffing Requirements 40
3.2 CIS Planning Milestones 43
5.1 Comparison of SDLC Approaches 80
5.2 A Functional Hierarchy 82

I. INTRODUCTION

We are in the midst of a revolution in management methods. Electronic data processing, quantitative analysis, management information systems (MIS) and decision support systems (DSS) are the revolution's tools of progress.

The computer is a challenge to the managers who must control the daily activities of many people. How should they manage in this environment of rapidly changing technology, expensive equipment and technical expertise? How can they efficiently and economically control the computer systems that are being designed for their organization's use? How can they predict the impact of future systems on their management control capabilities? Of equal importance is the question of how they can motivate the professional person who once made decisions alone, but now must interact with a computer. [Ref. 1: p. 15]

The extraordinary evolution of computer and communications technology has far exceeded our ability to plan and manage change in the information systems (IS) environment. These radically improved technologies provide end users with a powerful, direct link to sophisticated data processing systems being used to solve increasingly complex business problems. The term end user implies the ultimate user of the computer resource not an interim user such as a programmer, programming functions for the end users. During the past 30 years, some of the more remarkable advances have occurred in the area of "user friendly" systems development. These systems have effectively moved the computer from the organization's back rooms to become an integral part of business life. While this movement would seem to naturally draw computer professionals and end users closer together, the opposite often happens.

For their part, computer professionals have been slow to make the transition from technical supervisor to business manager. They have failed to develop management skills needed to plan, implement, and manage the introduction and use of computers in their organization. Instead of becoming masters of the new technology, they have sometimes become its unwitting victims. [Ref. 2: p. ix]

End users, impressed with vendor marketing hype, believe that computers can do almost anything. Armed with this misconception, they tend to flood their data processing (DP) department with application requests. Most requests are legitimate but are also later intensive projects. The typical DP department has a three-year backlog of development and maintenance work [Ref. 3: p. 96]. This backlog consists of more than just programming tasks. There's also work to be done in planning, analysis, design, evaluation, selection, training, documentation, implementation, maintenance, and conversion.

The backlog is a large part of the wall that separates data processing from the end users. To DP, the backlog is evidence that the department is overcommitted, understaffed and subject to insatiable demands. To end users, the backlog gives clear proof that data processing continues to take a larger bite of the organization's budget without being able to deliver on its promises. [Ref. 3: p. 96]

The key challenges in the eighties for computer professionals and end users will be to combine technical expertise with general business and management skills, to recognize the value of increased user participation in the development and operation of new computer systems, and to adopt structured development methodologies which can produce systems that are economical, efficient, and may be applied globally to the organization's business functions.

Civilian and military managers in the Department of the Navy have been beset with similar challenges. In contrast to their contemporaries in commercial enterprises, Navy managers are further constrained by a whole set of government regulations that complicate the acquisition of computer systems. In addition, the regular turnover of key military managers disrupts the continuity of the leadership involved in computer development efforts. Rarely will the military personnel who initiate a computer system project see it through to its completion. Consequently, major Navy computer system developments can span 5 to 10 years, or longer, and involve several different groups of military managers before the first end products are made available to users. The effect has been to retain many Navy computer systems well beyond the time when it is both practical and feasible to replace them with more advanced systems.

When legislative controls were initiated in 1965, they were meant to centralize and coordinate the acquisition of automatic data processing equipment (ADPE) for Federal agencies and to promote competition in the oligopolistic computer industry. Over the past twenty years, Congressional legislation has not kept pace with the dramatic technological improvements or the diversification of the computer marketplace. Processing power that once required a mainframe is now available on portable microcomputers which can be purchased at several retail department stores. End users have the capability to design their own applications utilizing sophisticated software packages. This disparity between current procurement laws and technological advances has provided resourceful Navy managers with an alternative to costly mainframes. New or upgraded computer systems can be acquired quickly when a small, relatively inexpensive computer will fit the users's informational needs and budget. The result has been a

resurgearce of the urccntrolled and incompatible growth of computer systems within the Navy.

It's tempting tc cite cutdated regulaticns as the principal limitation in the acquisitien of Navy ccmputer resources. This assertion would be partly right and partly wrong. It's right because the quantity and quality of hardware has increased while ccmputer equipment costs have decreased to the point where lcng-established expense limits become repressive. It's wrong, however, to suggest that the lawmakers were mycpic in their perception that ccmputer systems were difficult tc manage. That observatien holds true and perhaps it is more relevant in today's dynamic computer envircnment. The large assortment of technolcgies currently available offers Navy management many cpportunities to implement viable computer sclutiens or threw tcgether utterly disastrcus systems. The difference between these opposing results often depends on the accuracy and ccmpleteness of user specifications. If the users understand what they want and can define their needs clearly, the chances of delivering a successful system are substantially increased.

This thesis reviews scme of the technclcgies and management methods that can be applied to the develcpment of computer information systems within the Navy. Additicnally, this paper addresses many technical and human factcrs that influence the outcome of computer projects. No universal approach exists fcr planning all facets of infcrmation systems. Navy managers will have to select those techniques, methods and methodologies that suit their organizational mission and objectives, expertise levels, and resource constraints. Managers should expect to vary their set of development techniques from project to project. These management methods, in effect, can be used as a develcpment toolkit. They can help Navy managers plan,

design, and build comprehensive information systems within the prescribed acquisition guidelines.

This study begins with an overview of information systems and the regulations that inhibit their widespread development in the Navy. This author contends, however, that the lack of education and inadequate participation by user groups poses the most serious threat to information system developments. Strategic plans, accurate system specifications, and the introduction of new technological capabilities must be driven by end users. In order to achieve the most effective and efficient use of computer resources, users must be willing to learn the technical aspects of information systems development that once were the sole concern of computer professionals.

With this view, Chapter 2 addresses the types of information systems and many of the regulations that govern their acquisition and use within the Navy. Strategic information systems planning and its relationship to organizational planning is discussed in Chapter 3. Chapter 4 investigates two methods that can be used to analyze and develop a preliminary design for computer-based information systems. System development life cycles, development alternatives, and project management issues are reviewed in Chapter 5. Finally, Chapter 6 addresses user-oriented applications development software and how it can increase productivity within an organization.

II. INFORMATION SYSTEMS AND GOVERNMENT REGULATIONS

A. INTRODUCTION

Effective management depends on accurate, timely and reliable information. Modern computer systems have evolved in response to the diverse user needs for information. Commercial enterprises must have information to maximize profits and remain competitive. Government agencies need information to effectively and efficiently carry out their prescribed missions.

While information systems have flourished in the private sector, government agencies have witnessed the deterioration of computer resources that once were the leading edge of technology. Many Federal agencies continue to operate with computer equipment that was manufactured in the 1960s. One reason government agencies lag behind commercial entities is clearly the mountain of bureaucracy that restricts the timely acquisition of computer resources. Less clear, are the reasons why Federal agency management has not developed methodologies to effectively implement information systems in the shadow of government regulation. Perhaps the number of antiquated computer systems operating within Federal agencies reflects the obsolete management practices that have sustained them. While Congress was restricting government computer growth, businesses throughout America were experimenting with the computer's power and versatility.

The Federal government is beginning to wake up to the realization that its agencies possess inadequate information systems and agency managers lack the necessary experience to rapidly assimilate modern technologies into their

organizations. The Department of the Navy (DON) shares the burden of these problems with other Federal components. Fortunately, private enterprises have pioneered new approaches (and suffered the pain!) in the development of advanced information systems. Navy managers, particularly those with limited computer skills, must study the lessons provided by American businesses, learn them quickly, and proceed with the construction of viable information systems within their organizations.

Congress, through recent legislation, may have unknowingly commited Federal organizations to building the most sophisticated information systems in general use. It appears to be a time when Congress will accommodate state-of-the-art information system projects that are well specified and that engage the concepts of Information Resource Management (IRM). This chapter briefly reviews the components of an information system and the diverse regulations that make it difficult for the Navy to purchase computer resources while implying simultaneously that more progressive information systems are needed.

B. INFORMATION - A VITAL RESOURCE

Any organizational structure that implements a complex system is made up of parts that are interrelated and that function together. The interrelationships among the parts of the system lie in the sharing of the resources used. One resource that must be shared by viable systems is information.

Information is an essential resource for any functional system that delivers planned results. Therefore, any functional system, within any organization, should encompass methods and procedures for developing and delivering information. This is known as an information system. [Ref. 4: p. 9]

Information systems are formed through the coordinated functioning of people, equipment, procedures, data, and other resources to provide uniform, reliable, accurate information. An organizational system is tied together by its informational elements that permit the system to function cohesively. Because information is a universal tool for the operation of any organization, information systems tend to involve persons in multiple parts of an organization cutting across departmental boundaries. Information is a resource just as money, materials, facilities, and people are, and the use of this resource must be carefully planned and controlled with a variety of management techniques.

C. THREE LEVELS OF INFORMATION - THREE INFORMATION SYSTEMS

Distinct information needs exist at several organizational levels. Informational support is needed in controlling the daily operations of the organization, in ongoing management, and also in planning strategic changes for future years. Each of these levels of information need has evolved its own types of information delivery tools. [Ref. 4: pp. 9-10] To meet specific areas of management needs, three types of closely interrelated information processing systems have been implemented.

1. Electronic Data Processing (EDP)

Electronic data processing (EDP) establishes operational controls over the organization's routine activities and transactions. EDP was first applied to the lower levels of an organization to automate the paperwork. The basic features of EDP include:

1. A focus on data, storage, processing, and flows at the operational level
2. A system for efficient transaction processing

3. A need for scheduled and optimized runs
4. The use of integrated files for related jobs
5. The production of summary reports for management

The EDP level of information systems supports the functional subsystems of an organization. Emphasis is on recording basic operational details associated with the organization's daily transactions. EDP systems capture this basic operational data and generate the documents necessary to tie together all related functions during the normal conduct of the organization's activities. In addition, files created in the EDP system become the source of information for higher levels of managerial control and planning functions. Essentially, the EDP system establishes an information base for all integrated functions of an organization.

Technological advances such as increased hardware capacity and speed, on-line operating systems, enhanced data communication devices, and "intelligent" terminals made the EDP level of activity in many organizations an efficient production facility for transaction processing. The next evolutionary step was to focus on management concerns about integrating and planning for an aggregate of the organization's subsystems. The result of this effort was the development of management information systems.

2. Management Information Systems (MIS)

A management information system (MIS), basically, involves computer assisted procedures for reviewing the results of daily transactions and calling attention to situations that require special concern or decisions. These systems apply the power of computers to review information records on the basis of their data content. Managers establish the standards, or boundaries, that separate normal conditions from those requiring attention. The system may

then alert management to those exception conditions that require human intervention and decision making. [Ref. 4: p. 11]

Besides exception reporting, an MIS provides a resource for summarizing information about the status of the organization's activities. This capability helps managers derive meaningful information quickly and accurately for controlling the entire organization or any of its segments. Sprague [Ref. 5: p. 7] summarizes these elevated features of MIS data processing as having:

1. An information focus, aimed at middle managers
2. Structured information flows
3. Integration of EDP jobs by organizational function (e.g., administration, personnel, planning, etc.)
4. Inquiry and report generation (usually with a data base)

Thus, EDP systems provide detailed information, while management information systems provide selective information through further processing of detailed information. Although MIS contributed a new level of information processing to serve management needs, it was still oriented to, and built on, information flows and data files.

A third dimension of management is to envision the future structure and functions of the organization and to establish long-term plans to meet these goals. Decision support systems (DSS) evolved to assist managers in this planning dimension.

3. Decision Support Systems (DSS)

The DSS concept focuses on the highest level of the organization. It utilizes the results of EDP and management information systems and may include additional data brought in from external sources. DSS emphasizes features that

provide an interactive computer-based system to help decision makers solve less frequent, unstructured problems. Sprague [Ref. 5: p. 6] presents the characteristics of DSS as being:

1. Decision focused, aimed at the less well structured, underspecified problems that upper-level managers typically face.
2. An attempt to combine the use of models or analytic techniques with the traditional data access and retrieval techniques.
3. Specifically focused on features that make them easy to use by noncomputer people in an interactive mode.
4. An emphasis on flexibility and adaptability to accommodate changes in the environment and decision-making approach of the individual users.

By incorporating the organization's own data with external data, such as the state of the economy, demographics, and government policies, a DSS can, in effect, look ahead and project operating results based on the conditions and assumptions supplied by planners. The DSS becomes a tool for producing a model or simulation of the future state of the organization. [Ref. 4: p. 12] Viewed together, these three interrelated subsystems, EDP, MIS, and DSS, establish the framework of an overall systems capability known as a Computer Information System (CIS). The CIS is a total system that includes the use of computers and encompasses all computer related information processing within an organization. While the evolutionary growth of hardware and software tools for putting together a computer information system offers management a wide selection of alternatives, the phenomenal rate of growth of these tools creates numerous design and implementation problems.

D. COMPUTER GROWTH BECOMES A MANAGEMENT PROBLEM

EDP has been in use since the mid-1950s. MIS developments were introduced during the late 1960s. From the mid-1970s to the present, DSS development has been emphasized. Major new computer lines appeared about every eight years during the 1960s and 1970s; that cycle spins almost twice as fast now. [Ref. 6: p. 165]

The expansion and growth of technology has spurred the evolution of computer systems from the large mainframe unit to the departmental minicomputer and then to the office microcomputer. As technological developments accelerated and user demands multiplied, computers and office automation equipment were installed throughout many organizations. However, the widespread use of small decentralized computer systems posed difficult management problems when compared with centrally controlled mainframes. This has led to two basic views of how computer systems should be managed. Proponents of centralization argue that centralized computing ensures efficiency and permits effective service to all users. Proponents of decentralization say that distributing computer resources throughout an organization is more cost-effective and improves end user productivity. While there seems to be no agreement in the arrangement of computer systems, private enterprises are moving toward decentralized (distributed) systems but they are retaining centralized control over the planning, acquisition and use of computer resources.

As more versatile systems were developed, many commercial organizations discovered that there was only a limited capability of interaction between various types of computers. These organizations were trying to operate with unrelated and incompatible hardware and software. Because of increasing problems with data/information processing,

corporate managers recognized the critical nature of controlling the computer resource. They realized that management and control of the computer and the corporation's information resource had been neglected.

E. INFORMATION RESOURCE MANAGEMENT (IRM)

The multi-faceted nature of the information resource brought about the concept that a single function must be responsible for office automation, communications, and data processing. Since these technologies are interrelated, the concept of a single integrated plan and implementation schedule is viable and necessary for their maximum effectiveness. In addition, consideration was given to the level at which responsibilities were focused so that comprehensive systems plans closely tied to both corporate and unit business plans. This was done because many organizations realized that the information management function had been "buried" in financial or administrative service areas and that it more appropriately deserved its own area. Thus, the concept of information resource management (IRM) was adopted.

Information resource management helps an organization integrate business needs, personnel, hardware, software, communications, and office automation within the scope and financial resources of the enterprise. A basic premise of information resource management is the ability to make information available to whomever needs it when and where it is needed. The information resource environment must include a structure with the function of managing data/information. Many organizations are developing the function of Data Administration which has the managerial responsibility associated with planning and controlling of all data that is used throughout the enterprise. Data

administration has the objectives of maximizing the availability of data and exercising control of the data. This function acts as a liaison between top management, end users, and the DP department. As a result, the information systems plans developed within the data administration environment tend to have better user commitment as well as the solid appreciation of management.

F. REGULATIONS SLOW GOVERNMENT COMPUTER DEVELOPMENT

Although the private sector has been forging ahead with IRM practices, most Federal agencies are just now adopting similar concepts. Federal agencies lag behind private enterprises in computer systems development mainly due to legislation that was initiated twenty years ago.

1. The Brooks Act

The Brooks[1] Act, (Public Law 89-306) enacted 30 October 1965, established the basic framework for Federal computer applications. This legislation authorized and directed the General Services Administration (GSA) to coordinate and provide for the economic and efficient purchase, lease, and maintenance of automatic data processing equipment (ADPE) by Federal agencies. Two other agencies, the Office of Management and Budget (OMB) and the National Bureau of Standards (NBS), were also given significant authority over government-wide computer activities. OMB was tasked with overall policy guidance and to mediate disagreements between GSA and user groups while NBS was tasked with the development of Federal Information Processing Standards (FIPS). [Ref. 7: pp. 18-20]

---

[1]This legislation was sponsored by Representative Jack Brooks (D-Tex).

Before this time, Federal agencies purchased or leased ADPE based on individual needs, resulting in uncontrolled, large expenditures for computer resources. Many of the computer applications in the Federal government were unique. The size, scope, and complexity of these applications presented serious problems in areas such as planning, policy, design and acquisition. Congress noted these problems and quickly moved to control the proliferation of computer systems within the Federal government. The Brooks Act became the Congressional hammer to exert control over Federal ADP spending. This legislation was enacted before the emergence of software as a major portion of the cost of a computer system. Although the Brooks Act was specifically directed at hardware and hardware maintenance services, commercially available software is now considered to be included in its provisions.

The Brooks Act has given rise to a multitude of regulations governing Federal ADP acquisition and management. Executive regulations which have been published in response to the Brooks Act include Federal Property Management Regulations, Federal Procurement Regulations and GSA's Federal Management Circular 74-5; eight OMB Circulars; various reports and studies published by the General Accounting Office (GAO); and the 100-plus FIPS developed by NBS.

Within the Department of Defense (DOD) similar regulations governing ADP acquisition and management have been developed. DOD Directive 4105.00, "Selection and Acquisition of Automatic Data Processing Resources," and DOD Instruction 5100.40, "Responsibility for the Administration of the DOD Automatic Data Processing Program" are two key documents that control military ADP expenditures and operations. The Department of the Navy (DON) followed the DOD's lead by promulgating these policies within the Navy.

The Secretary of the Navy (SECNAV) has issued over 40 instructions, the most important of which is SECNAV Instruction 5236.1A, "Specification, Selection, and Acquisition of Automatic Data Processing Equipment" which establishes the guidelines, dollar approval thresholds and required documentation to support computer procurements within the Navy. At the next lower level in the Navy hierarchy, the Chief of Naval Operations (CNO) or OPNAV level has issued over 35 instructions governing the management of computer resources directed at all naval organizations.

As can be seen by the numbers of regulations at every level within the Federal and military system, the desire to encourage effective and efficient acquisition and management of ADP resources cannot be overstated. Federal agencies, in keeping with the spirit and intent of these laws, have experienced some debilitating side-effects. These rules have fostered a Federal ADP acquisition life cycle replete with lengthy justification requirements and interminable reviews. The result is that agencies have been effectively and efficiently blocked in their attempts to acquire more capable computer systems. In recognition of the newly emerging concept of IRM, the Federal government has further legislated controls in the Paperwork Reduction Act of 1980 and, more recently, in the promulgation of The Federal Information Resource Management Regulation (FIRMR).

2. Paperwork Reduction Act of 1980

The Paperwork Reduction Act of 1980 implies that Federal agencies have not used strategic planning in managing the computer resource. It addresses the subject of Information Resource Management by requiring each Federal agency to designate a single individual who is responsible for all agency information systems. Each official,

designated as an agency's information resource manager, reports directly to the agency head to carry out his IRM responsibilities. The IRM subsystems include, but are not limited to, data processing, records management, forms control and telecommunications technologies. This law, besides reducing paperwork and improving the efficiency of Federal information policymaking, mandated the preparation of a five year plan for data processing and telecommunications resources. [Ref. 7: p. 9]

Under the Paperwork Reduction Act of 1980, each Federal agency is responsible for carrying out information management activities in an efficient, effective and economical manner. To assist agency management, the Office of Information and Regulatory Affairs (OIRA) was created within OMB for developing and implementing Federal information policies, principles, standards, and guidelines that form the Government's information management policy. The Director of OIRA is tasked with the selective evaluation at least once every three years of the information management activities of each Federal agency to assess their adequacy and efficiency.

3. The Federal Information Resource Management Regulation

The Federal Information Resource Management Regulation (FIRMR) became effective 1 April 1984. This regulation provides a single directive concerning the effective management of automatic data processing, office automation, records management and telecommunications. Its emphasis is on managing information throughout the life cycle (from collection or creation to disposal). This regulation is intended to provide a logically organized guide to Information Resource Management for all Federal agencies. [Ref. 8: p. 20994]

The Paperwork Reduction Act of 1980 and the FIRMR introduce an ironic twist to the Government's historical ADP acquisition strategy. The requirements for planning and controlling the use of computer resources has been strengthened and extended. The Executive decision makers apparently can no longer resist the temptation to adopt and replicate the successful concepts of IRM developed by private enterprise. The Paperwork Reduction Act of 1980 specifically requires the use of advanced database development tools such as database management systems and data dictionary systems (these tools will be discussed further in Chapter 6). The OIRA has been created as a watchdog agency to help enforce the data/information management standards. The FIRMR implies that strategic planning and new management functions which include Data Administration and Database Administration, must be incorporated into an agency's organizational structure to comply with the law.

The meaning of the newer regulations is clear. Federal managers must view data/information as a resource and they must assume responsibility for its use within their respective organizations. These new requirements implicitly and explicitly call for sophisticated data management standards, procedures, and tools. Many of these requirements will be difficult or infeasible to implement on the Navy's older computer systems. Converting existing data so that it is useable with new technology will take years and be costly. If Congress and the other Executive managers are committed to the philosophy of IRM, then they must provide their Federal agencies with the appropriate ADP resources to do this job properly. The present rigid ADP acquisition life cycle must be streamlined to support IRM goals.

G. SUMMARY

The three-tiered structure is a practical approach to fulfilling an organization's information systems needs. All three information processing capabilities are rarely found within DON components. Some Navy organizations may not need all three capabilities. This author believes, however, that the total CIS environment is necessary for the majority of Navy organizations. The need for rapid, multiple information flows throughout the DON for the routine conduct of operations supports this contention. There are, of course, many other benefits with the total CIS approach.

How far behind private industry are the Navy's computer information systems? The answer to this question would probably entail reviewing a list of specific models of computers currently used in the Navy and then offering an estimate based on the oldest systems in use. This procedure would be inaccurate and meaningless. Should Navy managers use private industry as a measure of their system's capabilities? Definitely not. One lesson learned from industry is that organizations must develop information systems performance standards based on individual needs.

The plethora of regulations has certainly contributed to the obsolescence of the Navy's computers. Until new acquisition regulations are written, DON components will have to implement interim computer solutions (i.e., purchasing small computer and word processing systems). These interim systems, however, should be viewed as stop-gap measures and not be construed as an absolute means to deal with the status quo. It's easy for Navy managers to become cynical about computer acquisition after years under the stinging lash of Congress's tongue. The "new rules" mandate management action but are not a license to buy large quantities of computers without appropriate plans. Navy

management can only achieve their optimal information goals if they ardently pursue long-term systems planning and educate user groups in progressive development methods.

III. STRATEGIC INFORMATION SYSTEMS PLANNING

A. INTRODUCTION

Despite many years of experience with computers, data processing and non-data processing managers still face many unhappy surprises from their information system installations. These surprises frequently result from failures in long-range planning. Most organizations have adopted some type of strategic planning to implement organization-wide goals and objectives. However, the same principles have not been applied as well to CIS development efforts.

Computer professionals tend to concentrate on day-to-day trench warfare in a constant battle to deliver on the user's demands. This sense of urgency to meet today's operational requirements is understandable. Yet, we must also recognize that part of today's problems have resulted from a lack of adequate planning in earlier years.

B. PLANNING FOR CHANGE

Frequent changes in hardware and software technology, rapid personnel turnover, constant changes in systems requirements and the frequency of unexpected user demands are factors that contribute to the changing environment of computer information systems. The solution to dealing with these factors lies in setting a flexible strategic plan that will guide how these changes will occur. [Ref. 2: pp. 9-20] The following elements should be included in a long-range CIS plan:

1. Systems
2. Hardware
3. Software

4. Staffing
5. Control

Each planning element is developed as a separate topic within the overall CIS plan. Since interdependencies between elements will probablly exist, related items among the subplans must be cross-referenced. The total CIS plan is produced from the combination of the five elemental plans.

Before conducting any study of future information systems requirements, the existing computer resources must be reviewed and described in such a manner to provide a basis for establishing each part of the total plan. Descriptions of existing systems should summarize the types of applications being used; currently installed ADPE and telecommunications devices; the types and quantities of data files in use; daily, weekly, and monthly computer usage statistics based on data processing workload requirements; the number of programs and the types of programming languages in use; and any requirements for specialized software such as data base management systems, report generators, and telecommunication control software. The Systems and follow-on plans can then be developed from this summary information of current computer resources.

1. The Systems Plan

The systems plan requires development of a clear concept of how the various functions of the organization interrelate and how the systems currently in operation assist these functions. Information system managers must familiarize themselves with organizational and departmental plans, the organizational structure, the organization's business methods, and its products and services. Non-data processing managers must get involved in the planning process by contributing their experience and knowledge of business processes.

Developing the systems plan is probably the most time-consuming and critical portion of the long-range planning effort. Gaining the commitment and the voluntary participation from key managers for this task can be a major obstacle. The reward for executive level effort, though, is the potential for more responsive systems that meet management's specific informational needs.

The systems plan should contain two major categories which cover those functions that are directly supported by computers and the functions that are not computer supported. Fried [Ref. 2: pp. 11-12] states that the choice to automate a particular function can be determined by assessing the application based on the following information:

1. A review of potential changes of these functions with the responsible organizational units
2. An examination of the function for automation potential
3. An outline of the systems concept (a brief flowchart of the information process and five or fewer pages of narrative)
4. A review of the systems concept with potential users
5. A final technical system concept paper
6. A description of system resource requirements
7. An estimate of the computer resources necessary for development, testing, and converting the new applications

After all the above information has been collected and summarized, cost estimates are prepared for changes to the existing system and for anticipated systems. Current costs of operating the function, current and future capabilities of the system, and the economic impact on present labor-intensive methods are numerically evaluated. The resulting documentation should show the projected cost of current versus proposed methods over five years including

a payback analysis for each application in the plan. The combined cost and descriptive information will help to isolate potential changes or new applications that do not appear economically feasible and that are better served by noncomputer solutions.

Applications that are financially feasible, and that cannot be resolved without the use of a computer, must be reviewed with top-management. The selected applications should be examined for priority in terms of funds availability, payback period, consistency with the organization's long-term business plans, and anticipated (business-related) environmental conditions. [Ref. 2: p. 12]

Fried, Powers, et. al. [Refs. 2,4 p. 12, 30] agree that the most productive approach in the final review and selection of CIS proposals is to establish a steering committee. The steering committee is composed of top-level management personnel representing all user areas. The chief executive, or head of the organization, should chair this committee providing the leadership, authority, and commitment that major CIS investments require. The responsibilities of the steering committee are to approve the long-range CIS proposals, approve individual segments of the proposals and establish the priorities of the approved applications. A further responsibility of the steering committee will be to periodically monitor the progress of approved systems to ensure that design and cost constraints are within established limits.

The documentation from this proposal/approval process becomes the organization's long-range systems plan. On completion of the basic systems plan, three related shorter duration plans which address hardware, software, and staffing requirements should be developed concurrently to implement the systems plan objectives.

2. The Hardware Plan

Information from current computer operation reports, combined with projected volumes for present systems and on proposed systems, provides the basis for forecasting hardware requirements. The hardware plan should include a year-by-year statement of capacities, capabilities, locations, costs and methods of transition from present configurations to future ones.

Since the planned applications represent an extension or replacement of the current work load, a summary of the data shown on the descriptions of present applications must be integrated with the expected additional work load of planned applications and development work. Estimates should be made in terms of the performance of the current hardware. For example, total anticipated main memory and peripheral unit needs should be estimated on the basis of the needs of the systems that are currently, or are expected to be, operating concurrently in a multiprogramming[2] mode.

Having established the technical specifications, the next step is hardware evaluation. This task includes technical evaluation and possible benchmarking[3] of equipment, single- or multiple-vendor support, and procurement options such as buy, lease or rent. Of particular importance in this evaluation process are two factors that affect hardware economics: the rapid gains in technological improvements and lower costs associated with new equipment relative to older systems.

---

[2]Multiprogramming refers to the process of overlapping and interleaving the computations of more than one program to maximize the use of the hardware and software resources of the computer system.

[3]Benchmarks are standardized computer programs used to test the processing power of different computers. They are one way by which machine characteristics can be compared regardless of programming language or hardware construction.

The shortened life cycle of systems means that buyers must study trends in hardware and software to avoid acquiring equipment that is near obsolescence. The exception to this guideline is that it may be justifiable to acquire a used computer near the end of its life cycle to realize substantial cost savings. The primary limitation with older models is that costs for technical support are likely to increase as the system approaches retirement. These costs can become exorbitant, particularly when a vendor discontinues a line. Even if the computer is provided free, maintaining it, in some cases, can be an uneconomical venture. [Ref. 6: pp. 165-166]

Another consideration is that vendors have recognized the shortened life cycle of systems and tightened leasing arrangements accordingly. They are charging a premium for shorter-term leases of three to four years compared to the traditional seven. For buyers, the primary recourse means finding those vendors whose computers are compatible with their organization's encumbent systems and are likely to be compatible with future generations of hardware [Ref. 6: p. 166].

Hardware selection cannot be done without considering available software options and the staffing level consistent with authorized expenses. The schedule for implementing the hardware changes depends on the priorities set forth in the systems plan and incorporates the staffing and software plan requirements for development and continued operation of the applications.

3. The Software Plan

In the early years of computing, people operated the computer system. Programs were loaded and extracted, data was input, and computational results were generated by manual intervention with the computer and its associated

devices. Software, collections of interrelated computer programs, have displaced humans in performing these functions. The term "operating system" refers to a specific set of programs that have replaced the people who formally supervised and operated the computer. With modern computer systems, expanded capabilities are also controlled by software. The types of software vary with the specialized requirements of many functions that are performed during the routine use of the computer. Systems software, therefore, must be selected according to how it will be used in the control, monitoring, development and management of computer resources.

Software can be classified according to its use in application, development, and operating requirements. Applications requirements encompass software that controls the execution or manipulation of data by end users. These programs are designed to monitor data communications; control terminal/user interaction with the system; permit data to be extracted from or inserted into a data base; and allow users to query the system and generate summary reports. Development requirements software are the set of programs normally used by data processing personnel to create and maintain application programs and databases for end users. Development software includes all applications software plus those programs necessary for the standardization and cataloging of data items, files, and programs; updating and documenting of application programs; facilitating on-line interaction with computer resources; and software to monitor and detect errors in applications programs. Finally, operating requirements software are the set of programs used to oversee the routine use of computer resources. This type of software includes programs that keep track of application program and magnetic tape libraries; monitor and analyze the performance of computer

hardware; and account for computer resources used during system operation. For a distributed[4] environment, a similar list must be drawn up for minicomputers, microcomputers, and any network control software.

The software plan, like the hardware plan, is developed according to the timetable specified in the systems plan. Software selection will influence and be influenced by manpower and hardware requirements. Introducing a new, more efficient operating system for instance, may affect follow-on hardware selection, documentation and technical standards, staffing levels and user training.

Other considerations that must be addressed by the software plan include the anticipated price of the software; whether to develop programs in-house, modify an off-the-shelf package, or purchase a custom package from an outside vendor; anticipated costs of conversions; and other costs associated with software maintenance, enhancement, and the updating of technical documentation. [Ref. 2: p. 16] Selecting the proper mix of hardware and software is critical to the systems development effort. A third area, staffing, will also have a major impact on the implementation of new applications.

4. The Staffing Plan

The selection of hardware and software systems will designate the specialized computer skills required to meet the systems plan objectives. Within limits, routine perusal of currently published materials will provide an adequate indication of general trends in computer professionals' capabilities and corresponding salaries. Various computer

---

[4]A distributed processing system is characterized as having both the processor and data storage facilities physically dispersed and interconnected by data communications facilities.

magazines report on current salaries and other benefits that computer professionals seek. The article, "Salary-status Survey, Part I: Where the Dollars Are," Computer Decisions [Ref. 9], compares the average salaries and fringe benefits for computer professionals throughout the United States. Other sources of statistics on computer personnel salaries can be obtained from annual industry surveys such as: Source EDP Department SN, P.O. Box 7100, Mountain View, CA 94039; or Women in Information Processing Survey, Lock Box 39173, Washington, DC 20016. Anticipated salaries should be documented in the staffing plan as well as the costs for outside consultants or temporary employees when necessary.

The staffing plan should project specific manpower requirements for 18 months and show general projections for at least another 12 months (see Figure 3.1) [Ref. 2: p. 17]. A training program (and its anticipated costs) should also be included for the continued development of personnel resources.

Because the CIS environment is a people-designed and people-controlled effort, the ability of the organization to project and meet staffing requirements will contribute to systems that are on time and within budget. Technical competence and experience are critical prerequisites to a well rounded DP staff. Good communication skills, however, are essential for those people who are expected to routinely interact and guide users in the use of computer resources.

5. The Control Plan

The first four plans that have been discussed will help managers organize the information concerning present and future computer resource requirements. The fifth plan is important because it assists management in controlling the areas of operations, development, maintenance, and the user interaction with the information system. Some

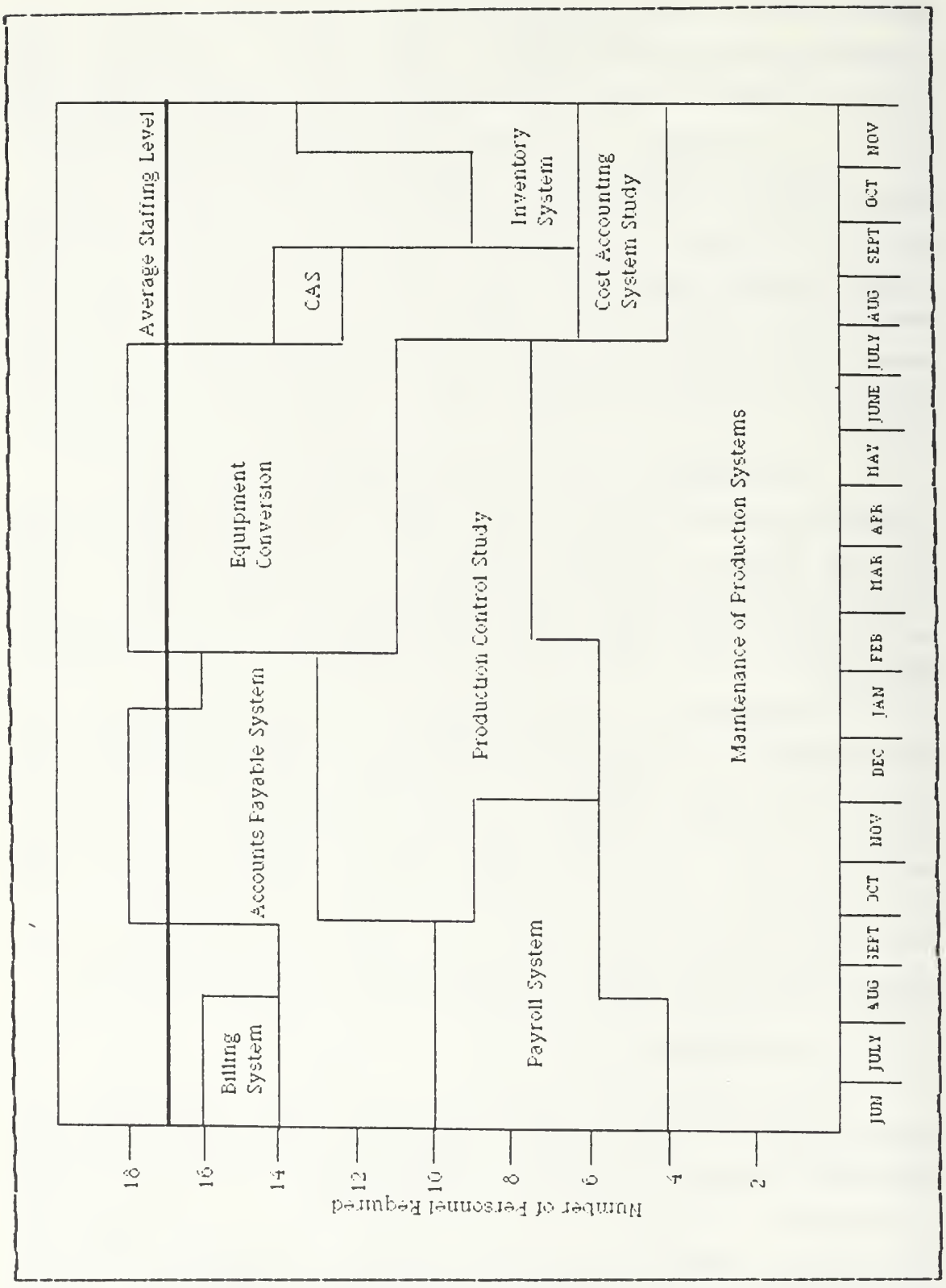

Figure 3.1 Staffing Requirements

organizations approach performance controls with the philosophy of minimizing the cost of information systems. Others pursue maximizing the benefits of information systems with considerably less emphasis on costs. With modern computer systems, the latter approach may be more appropriate because tangible benefits from acquiring sophisticated hardware and software can be marginal compared with the initial large capital outlays. Long-established productivity indicators may not be relevant to newer systems operations. Performance measures, therefore, should be continually reviewed and updated for the critical evaluation of advanced systems.

The control plan incorporates the policies, procedures and techniques necessary to provide management with the tools to monitor the performance and control the direction of system operations. After introducing a new application, management must ensure that the system is being operated properly, performs up to expectations, remains cost-effective and can adapt to changing conditions. During periodic project reviews, the steering committee will want summary progress reports on CIS operations to support go/no-go decisions on continued investment in the applications. Good management control depends on quality reporting. Fried [Ref. 2: pp. 16-18] suggests that the reports should:

1. Evaluate by measuring actual performance against a predetermined standard
2. Be oriented to the function being measured
3. Cover all functions
4. Chart a 13-month period to indicate trends
5. Predict trends
6. Enable management to anticipate potential problems or unusual expenses

7. Be concise and readable and interpret (graphic in presentation when possible
8. Support structural continuity from the lowest level of the organization to top management
9. Be received by management routinely and promptly enough to permit timely corrective action

Detailed chargeback reports must be established either for services rendered to the organization by an outside DP center, or for services provided by in-house CIS resources. It is essential to good management control that users be made aware and accountable for all costs of development, operation and overhead associated with their applications. Nolan [Ref. 10: pp. 114-124] suggests a chargeout system based on data output, such as the number of reports, schedules or invoices processed. End users understand and can help to control these "workload units" more easily than the usual computer-related measures of central processing unit (CPU) or main memory time.

Figure 3.2 summarizes the major milestones in developing a long-term CIS plan. Depending on the size of the organization, the scope of the plan, management commitment and available resources, it may take several weeks to perhaps a year to develop the strategic plan. [Ref. 2: p. 19]

C. AVOIDING FAILURE

Strategic planning, when done properly, has the tendency to stand an organization on its head. That is to say, the process is normally approached from a top-down perspective but its successful implementation relies heavily on support from the organization's lower levels. Internal personnel resistance will thwart the most carefully laid plans.

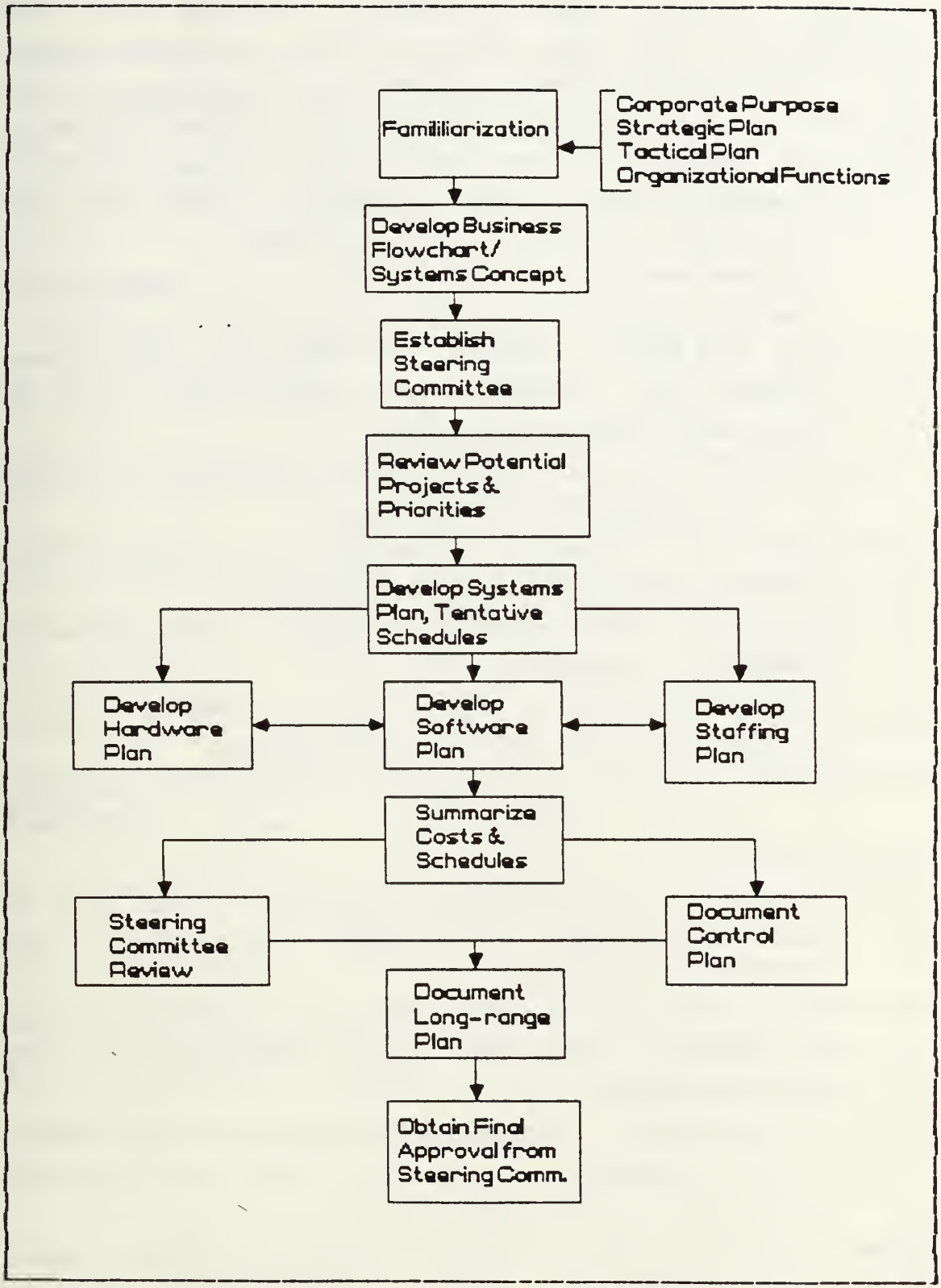

Figure 3.2 CIS Planning Milestones

Tichy [Ref. 11: pp. 204-206] contends that many companies have done a poor job of strategic planning because they treated it as a gimmick rather than a central aspect of management. He refers to 10 pitfalls of strategic planning which were identified in the early 1970s. The majority of companies which tried strategic planning during that era stumbled over one or more of these problems:

1. Top management's assumption that it can delegate the planning function to a planner (or planning group).
2. Top management becomes too engrossed in current problems and doesn't spend sufficient time on long-range strategic problems.
3. Failure to develop company goals suitable as a basis for formulating long-range plans
4. Failure to assume the necessary involvement in the planning process of major line personnel.
5. Failure to use plans as standards for measuring managerial performance.
6. Failure to create a climate in the company which is congenial and not resistant to planning.
7. Assuming that the organization comprehensive planning is something separate from the entire management process.
8. Injecting so much formality into the system that it lacks flexibility, looseness, simplicity, and restricts creativity.
9. Failure of top management to review with departmental and divisional heads the long-range plans which they have developed.
10. Top management's consistent rejection of the formal planning mechanism by making intuitive decisions which conflict with formal plans.

The most telling aspect of Tichy's forecast is that nearly all of these errors boil down to an ability to deal with

people, a critical, if not the most critical, management responsibility. The 1980s are as uncertain and subject to major technical transformations as were the 1970s. Strategic planning and decision making will continue to take on an increasingly important role. It appears to be a time when organizations will need to learn to do it right.

D. SUMMARY

Strategic IS planning requires a broad mix of studies and evaluation methods. The existing computer work outputs and capabilities must be assessed in relation to current and projected organizational activities. Present and future work activity levels must be evaluated in terms of feasibility for automation and to the extent that automation is necessary. A logical (user view) design of the system must be produced for the computer specialists to translate into a detailed specification design. Implementing the results of the various studies, user specifications and detailed technical designs requires subdividing the overall information objectives into activity phases with discernable milestones.

Few individuals (if any) within an organization possess the prerequisite skills to accomplish IS strategic planning on their own. The blend of appropriate disciples must come from a combination of functional and DP management. The inherent complexity in the planning and design activities and the mechanisms to integrate project teams calls for formal procedures. Several of these management issues will be addressed in the following chapters.

IV. ANALYZING THE PRESENT-PROJECTING THE FUTURE

A. INTRODUCTION

The planning framework presented in Chapter 3 provides guidelines for the types of tasks and documentation required to set long-term CIS goals. The goals of information systems development should go hand-in-glove with the overall business objectives and goals of the organization. Frequently, an organization's future states are driven by external influence from governmental regulations or changes in societal attitudes. Change may also stem from internal pressure of employee's concerns about upgrading working conditions or management's effort to improve the quality of the organization's products and services. The type of information system that an organization develops is influenced by these changes. Conversely, a new information system can change the internal operation and structure of an organization. Managers must be aware of change within their organization and anticipate any consequences that affect information system development.

Beckhard and Harris [Ref. 12: pp. 16-19] identify two essential conditions for any change effort to be effectively managed. First, the organization leadership must be aware of the need for change and of their response to changes or lack of response that has significant consequences. The second condition is that leadership must have a relatively clear idea of the desired end state. Thus, the prerequisites for setting a plan for change should include: a good diagnosis of the conditions causing a need for change; a relatively explicit description of the desired end state; and a clear and accurate assessment of the dynamics

of the present system. Appendix A contains a list of questions that can assist managers in evaluating several areas of DP support.

This chapter will concentrate on two methods for analyzing organizational processes, assessing the need for change, and how managers might go about developing a computer-based solution to them. The first method involves the use of IBM's Business Systems Planning (BSP) study and how it was applied at Fort Ord, a U.S. Army base located in Monterey, California. The second approach presents the major activities invclved in conducting a structured systems analysis for the initial investigation and feasibility study of user requested applications. Structured systems analysis (SSA), or systems analysis, is a partial methodology. SSA includes top-down problem decomposition, use of graphical languages, and model building as a means of communicating with users. It is beyond the scope of this paper to provide a detailed description of BSP or SSA. Rather these techniques will be reviewed in context with what managers can expect to derive from their use. DeMarco [Ref. 13], Dickover [Ref. 14], Ross [Ref. 15], and Teichroew [Ref. 16] provide excellent discussions of several structured techniques that can be applied to information system developments.

B. HISTORY OF BUSINESS SYSTEMS PLANNING

During the 1960s, managers at IBM (International Business Machines Corporation) realized that they had established little control and planning in the overall direction of internal information resources. Little coordination took place among divisions and organizational units. Each manufacturing plant and marketing region had developed and operated its own information system.

Consequently, individual systems carried out redundant functions but differed in design and performance so they could not be used interchangeably and could not communicate with each other. The result was an excessive drain on data processing resources while minimizing IBM's return on investment because the organization-wide information needs were not being accommodated. [Ref. 17: p.2]

In 1966 IBM took the first step in solving this problem by creating a company-wide Information Systems Control and Planning (ISC & P) Department. The group then set out to inventory and profile the existing business systems and IBM's plans for the future. Recognizing that their efforts must be directed toward satisfying business needs and not solely toward individual functions, planners established a set of information system strategies covering five major areas [Ref. 17: p. 2] :

1. **Fixed data responsibility.** Policies should be established that fixes the responsibility and accountability for data accuracy, consistency, and timeliness to a specific individual or group within the organization.

2. **Single source and parallel distribution of data.** Data should be centrally controlled and managed throughout an organization and throughout the data resource life cycle which entails acquisition, storage, access and disposition. Although centrally controlled, the data must be valid, timely, and shared among diverse user groups.

3. **Central control and planning of information systems.** Information systems should match the needs of all levels of management and support the organization's business objectives. This can be accomplished by

consistently designing and controlling information systems from a top-management perspective.

4. **Organizational independence of data.** Data must be processable by one or more applications and used by several different organizational subsystems. The best approach to data independence is to develop data base systems as an integral part of information systems.

5. **Resource sharing of data, equipment, and communications.** Resources used in information systems should be standardized and compatible with each other to maximize their effective use and to realize economies of scale.

Combining their knowledge of existing DP operations and the direction established through the set of strategies, the ISC & P department defined an integrated set of information systems. During the definition and design stages for these systems, many of IBM's customers showed interest in the then-new planning concept. IBM responded to their requests by establishing the Business Systems Planning (BSP) program in 1970. Since its inception, IBM's Business Systems Planning methodology has helped many organizations, public and private, to formulate their information systems plans toward the improved use of data processing resources and control mechanisms.

C. BSP OBJECTIVES

The main objective when conducting the BSP study is to develop an information systems plan that supports the organization's short- and long-term information needs. According to the BSP Guide [Ref. 17: p. 3] there are six other important objectives that help justify and clarify the approach:

1. Guide management, through the use of a formal, objective method, toward establishing information system priorities without regard to provincial interests. Information systems can be an integral part of an organization, critical to its overall effectiveness, and represent a major investment of time and money. Non-DP managers must agree on the orderly development of information subsystems that serve the most pressing needs of the entire organization.

2. Develop viable systems based on the business processes that are generally unaffected by organizational changes. The types and characteristics of data used in an organization do not change often. The values associated with data items, however, are constantly changing. A well designed information system depends on correctly identifying and structuring the data so that it can be used with the necessary flexibility.

3. Allocate the data processing resources for the most effective and efficient support of the organization's goals. Organizations are constrained by the amount of resources that can be dedicated to computer systems. The information system must be designed to maximize the benefits to organizational members in a cost-effective manner.

4. Boost executive confidence that sound investments in major information systems will result.

5. Provide systems that are responsive to user requirements and priorities.

6. **Identify data as a resource that should be planned, managed and controlled to be used effectively by everyone.**

D. KEY ACTIVITIES IN THE BSP METHODOLOGY

To successfully achieve the objectives identified in the preceding section, the BSP program is logically divided into thirteen major events. The first two are activities that involve preparatory tasks to set up the BSP study and the next eleven activities are the study itself. None of these activities can be omitted, as stressed in the BSP guide [Ref. 17: p. 10], but may be carried out in varying degrees depending on the users' familiarity with the BSP approach. The following major activity descriptions outline the BSP study approach.

1. Gaining the Commitment

One of the underlying concepts in the BSP method is top-down analysis with bottom-up implementation. To achieve meaningful results, the study must reflect the business views of top-level management. More important, one senior executive should be selected as the team leader who will work full time in the study and direct team activities.

Because approval of the study recommendations represents a long-term investment in the use of data processing resources, high-level planners must agree on the study's direction, objectives, scope and expected deliverables. For these reasons, top-executive commitment is a critical factor that sets the tone throughout the study.

2. Preparing for the Study

Before the study begins, preparations are made to orient team members and participants toward the goals of the study. Team members (4-7 functional managers including the head of information systems) may take a 3 and 1/2 day BSP Indoctrination course provided by IBM. Executive participants should be briefed on scheduled interviews, the study's work plan, checkpoint reviews and a preliminary outline of the final report from the study.

A control room is established to insulate team members from the usual work day interruptions. This room will be the team's designated working area during the six to eight weeks required for the study. The final step in this stage is a sponsor's review (usually the top executive) of all preparations with the team leader.

3. Starting the Study

The BSP study begins with a business review consisting of three presentations to team members. The sponsor first reiterates the objectives, expected outputs (deliverables) and perspective of the study relative to other organizational objectives and activities. The second presentation is conducted by the team leader who reviews the business facts that have been gathered, addresses political and other sensitive issues, and covers the decision process, organizational functions, key people, major problems and the users' image of the data processing department. The third presentation is an overview of the DP department by the Information Systems Director or one of his principal assistants. Topics include historical data concerning projects started in the last two years, current activities and major problems, and projections of planned system changes.

The intent of these presentations is to give the team an overall understanding of the business and of the present and planned data processing support.

4. Defining Business Processes

Team members must identify and describe the business processes before fcllow-on activities can be conducted. Business processes are defined as groups of logically related activities and decisions required to manage the resources of the business [Ref. 17: p. 29].

Emphasis in the BSP is normally placed on those processes necessary to manage the key resources. Each resource of an organization can be thought of as having a life cycle made up of several stages. A product life cycle, for example, has four stages: requirements, acquisition, stewardship, and retirement. The length of the life cycle can vary greatly with the particular product area but it is of no consequence in this approach. Business processes can be identified to describe the major activities performed and decisions made by the organization while managing the resource throughout its life cycle.

More important than understanding in which life cycle stage a given process appears, the team should concentrate their efforts on identifying the processes, eliminating redundant processes and highlighting those processes that are key to the success of the business.

5. Defining Business Data

Things that are significant to the business, termed entities, are identified by the team. An entity is a person, place, thing, event or concept. Data about these entities is grouped into logically related categories known as data classes. This classification is essential in helping the organization develop data bases with a minimum

of redundancy and that allow systems to be added without major revisions to the data base.

6. Defining the Information Architecture

The information architecture is a matrix formed by listing the processes along one axis and the data classes along the other. The relationship of business processes to data classes can be established by marking each point of intercept on the matrix with the letter "c" (where a process creates a particular data class) or by the letter "u" (where a process uses the the data in that category). This activity is done to ensure that all needed processes and data classes have been identified and that one and only one process creates each data class. The resulting graphic is a valuable communication tool. It is, in effect, a blueprint of the team's recommendations for long-range information systems implementations.

7. Analyzing Current Information Support

During this activity, the study team analyzes existing data processing support and develops recommendations for further action. Specifically, team members will examine the present organizational structure, information system applications, business processes, and data files to identify voids and redundancies. This analysis helps to clarify functional responsibilities and systems interfaces.

The team also produces a process/organization matrix which indicates: key decision makers; the management personnel having major and minor involvement with a process; and the areas currently supported by data processing. This event helps the team identify the individuals that should be interviewed.

8. Interviewing Executives

Executive interviews are vital to the success of the BSP study. They provide essential facts about operational requirements and interrelationships among the organization's activities. They also help to promote the cross-fertilization of management ideas and practices throughout the enterprise.

Executive interviews are conducted to validate the information gathered and analyzed in the preceding activities. Executive participation helps to substantiate objectives, problems, information needs and the value of information systems from the vantage point of the managers who use them. Notes taken during the interviews are used to update the matrices and other study materials.

9. Defining Findings and Conclusions

One of the principal tasks in this step is to identify those problems that require computer-oriented solutions and those that do not.

Business problems noted over the course of the study are analyzed and related to the business processes. Team members divide the problems into categories, draw up findings and conclusions about them, and document recommendations for setting priorities among the information architecture subsystems.

10. Determining the Architecture Priorities

Development and implementation should begin after the findings and conclusions have been reviewed with management. The team should assist management in selecting the lead applications, subsystems, and data base. The BSP Guide [Ref. 17: pp. 64-65] groups the major selection criteria into four categories:

1. potential benefits
2. impact on the business
3. probability of success
4. end user demand

Prospective applications can then be ranked for each of the four categories. The application's scores in each category are summed, and the total score for each subsystem can be compared against the other prospective applications. Thus, the application with the highest overall score is given top priority. The other prospective applications are ordered in sequence corresponding to their scores. This list sets the priority for implementing the subsystems identified in the information architecture.

Changes in the business environment may cause changes in development priorities. After each subsystem is implemented, remaining applications should be reassessed to ensure that they are in proper sequence. A related problem centers on recognizing that some subsystems are built on others. Thus, prerequisite systems will have to be developed before other, higher priority applications can proceed.

11. Reviewing Information Resource Management (IRM)

The BSP-developed plan can fail without proper controls. The concepts and principles of information resources management (IRM), the ability to make information available to whomever needs it when and where it is needed, are examined in context with the organization's existing information services.

The study team should address problems with the information resource management function. They may recommend changes to increase its effectiveness through establishment of a steering committee, incorporation of project control systems in development efforts, and establishment of the data administration function.

12. Developing Recommendations

Specific recommendations are drawn up to assist management in its decisions regarding follow-on activities. The key recommendation focuses on acceptance of the information architecture as the base for directing near- and long-term information systems planning. Other recommendations may include enhancing the information resource management function and increasing support for end user computing. For each recommendation there may be an associated action plan identifying key decision points and activities required to implement a project.

The collective documentation, namely, the information architecture, architecture priority list, and recommendations, form the strategic information systems plan for the organization.

13. Reporting Results

Completion of the BSP study is marked by the submission of a formal written summary and an executive presentation of the study's findings and recommendations. The purpose of the report and presentation is to further executive commitment for implementing the study's recommendations and to secure approval for the overall strategic information systems plan.

E. APPLYING BSP AT FORT ORD

1. Background

Fort Ord is a U.S. Army installation located 7 miles north of Monterey, California. It is the home of the 7th Infantry Division and provides facilities for the training and education of various Army units. Two sub-installations; the Presidio of Monterey (Defense Language Institute)

located in downtown Monterey, and Fort Hunter Liggett (a 166,553-acre reservation used for field training) located approximately 80 miles south of Monterey, are part of the Fort Ord complex.

Fort Ord also has support responsibilities for the Army Reserve. This area of responsibility encompasses the southern 18 counties of the state of California, ranging from just north of Fort Ord and as far south as the California/Mexico border. To coordinate this support function, Fort Ord has an Area Support Detachment at the Los Alamitos Armed Forces Reserve Center (near Los Angeles).

The main installation at Fort Ord serves a population of approximately 16,000 military, 2,800 civilian employees, 11,400 family members, and 46,200 retired personnel and their families. The mission of Fort Ord is to support the 7th Infantry Division, sub-installations, reserve components, and the military community in the Fort Ord areas of responsibility; to plan for mobilization, deployment and other contingency missions; and to enhance community relations and the quality of life. [Ref. 18: p. 2-1]

2. The Need for Change

In August 1982, installation of two IBM 4331 computers at Fort Ord was completed. These units replaced a variety of IBM computer systems manufactured in the 1960s. Fort Ord's Automation Management Office (AMO) had the responsibility for managing this transition and for continued operation of the systems.

Only minor problems were encountered in training the AMO staff on the new systems and user satisfaction increased sharply. The new systems provided both improved batch processing equipment and an increased capacity to handle interactive computing. With the new systems installation

behind them, the AMO staff and Fort Ord's planners made an assessment of current base operations, existing data processing support and the future direction of information systems development on the base.

Fort Ord's management reviewed those issues, internal and external to the installation, that would influence the planning for information systems growth. Internally, they found that:

1. managers had access to large quantities of data but little information
2. individual units within the organization were acquiring computer word processing systems without planning for maintenance, training or technical support
3. computer systems expenses were soaring
4. no plan to integrate systems existed
5. no priorities were set for automating units within the installation.

External concerns focused on budgetary and legislative constraints. Congressionally mandated controls require Department of Defense (DOD) components to accurately project future needs (usually 3 years into the future) for Automated Data Processing Equipment (ADPE). Other congressional controls include spending reductions on ADPE and barring the use of lease options. Within the Department of the Army, budget administrators further constrained the acquisition process by switching the category of funds which ADPE could be drawn against from the Operations and Maintenance Appropriation to Other Procurement Appropriation. Due to lower dollar thresholds under the Other Procurement rules, this fundamental change makes the purchase of most ADPE, including microcomputer systems, more complicated. Additionally, Army budget administrators failed to clarify the funding change, leaving it to lower

echelon components to determine how to allocate the necessary money for ADPE without violating existing laws. Faced with these challenges and lacking a comprehensive plan to deal with them, Fort Ord's leadership decided to conduct IBM's Business Systems Planning study.

3. The Study

The ISP study (Fort Ord's managers renamed it Information Systems Planning to express a more universal perspective) was accomplished from 7 November to 16 December 1983. Mr. Karl Keeler, a principal assistant to the Director of the AMO, related the following unofficial reactions and experiences in a presentation of the study to Computer Technology students at the U.S. Naval Postgraduate School.

The first step was to get the installation's Commanding General to approve the BSP study. The difficult task was not to get the commitment to do the study and to involve the heads from all directorates, "When the Deputy Installation Commander learned that these directors would be removed from circulation for 6-7 weeks," as Mr. Keeler put it, "He said we were crazy."

The Deputy Installation Commander wasn't the only person who questioned this approach. In the AMO itself, staff members wondered about conducting any systems study while restricting input from data processing specialists. "We (the study's planners) discussed how the input must come from those people who know little or nothing about DP," Mr. Keeler said, "and the data processing people thought that this was strange." The AMO director pressed on and was able to convince Fort Ord's leaders that the benefits produced by the study would outweigh any perceived risk.

Team members were selected and sent off to IBM's BSP Indoctrination course in Los Angeles, California. When they

returned, preparations had been made to equip a separate building to conduct the study. The planners wanted to put the team to work immediately so that they "wouldn't lose the knowledge and enthusiasm they had gained during the BSP course.

The study team gathered together in the specially outfited building, held the necessary pre-study "kick-off" briefings and then spent the next two days determining the "pecking order" of the group. This experience became one of the first lessons learned according to Mr. Keeler, "you just don't join people who have set political relationships and then expect things to go smoothly."

Although the study team had been educated in the BSP activities and the associated tasks, the first week of the study was spent organizing the thinking-process and reviewing information about Fort Ord's base operations trying to find a direction. Mr. Keeler explained, "The team began to develop multiple branches of thought about what the base processes involved, several of them were wrong and didn't lead to anything, so we called in an IBM consultant who did an excellent job of resolving these problem areas."

The study progressed well after the first week. Using the BSP methodology and through 42 interviews of key managers from all user groups, the team identified 200 areas that potentially required IS support. Later in the study, only 25 percent of these 200 problems identified were considered for automation. The other 75 percent would be analyzed and addressed separately through other ongoing management procedures.

The study closed with the executive presentation of the proposed information architecture and recommended follow-on action plan. The results were well received and adopted as a long-range IS plan for Fort Ord. DP specialists from the AMO staff were then assigned the task

of taking the information architecture and designing the data specifications. In January 1985, data specifications were completed for the lead projects - an installation data base and data base management system, and a local area network that would eventually be linked to Army Regional Data Centers.

Overall, the study had been a positive experience that produced both a flexible strategic IS plan and significantly improved communications between data processing personnel and user groups. Using the plan, Fort Ord's managers have projected, over the next seven years, the type and quantity of ADPE and related IS support compatible with the organization's informational needs. Additionally, they are better prepared to deal with the DOD planning, programming and budgeting process in the area of information systems acquisition.

F. PLANNING CHANGE USING SYSTEMS ANALYSIS

The development of computer information systems is a form of problem solving. The problem is to provide the right information, to the right person, in the right form at the right time. Usually, this problem is too complex to be solved in its entirety by any single individual.

The solution will probably entail many different computer programs, hundreds or thousands of individual tasks, processing several streams of input data and producing a number of forms of output and feedback. All of these functions must be integrated along with control and adjustment functions. This level of complexity requires a systematic approach to the development of computer information systems. [Ref. 4: pp. 18-20]

The systems approach begins with a top-down perspective of identifying and viewing the complex, interrelated

functions as integral elements of systems. Total system requirements are defined and then broken down into subrequirements of increasing detail. Although there is concern for the individual parts, emphasis is placed on the integration of components that produce the end products of the entire system. Because components are viewed as parts of an integrated whole, the total systems approach is an effective means for analyzing and developing solutions to CIS problems. [Ref. 19: pp. 112-113]

G. THE SYSTEMS ANALYSIS APPROACH

Systems analysis is the application of the systems approach to the study and solution of problems. Within a CIS environment, systems analysis can be applied to business problems that require development of computer information systems. The systems analysis approach makes it possible to understand problems and to shape solutions.

The systems analysis process involves seeing the business organization itself as a system, analyzing its goals and objectives, and understanding uses for the information that will be the end product of the problem solution. Viewing the problem from the perspective of the user of information is a primary focus of systems analysis. [Ref. 20: pp. 160-161]

In contrast to the non-DP thrust of IBM's BSP study, systems analysis provides a set of strategies and techniques for partitioning complex problems into various levels of abstraction. Graphic and narrative tools have been specifically devised to support this process and to systematically document its approach. Because the analysis and application of these tools can be confusing to untutored users, a systems analyst is used as a facilitater. [Ref. 4: pp. 22-23]

The systems analyst is a problem solving specialist who can help users to communicate their perspective of information processing needs and relate those needs directly to the design and development of computer-based solutions. The importance of the analyst's communication abilities cannot be overemphasized. Users and technical designers must understand each other to achieve the development objectives.

Before launching any in-depth development study, it makes sense to first validate user requests to improve or enhance existing systems and to explicitly define the problem. A list of questions developed by Wen [Ref. 21] that should guide the systems analysis process is contained in Appendix B.

1. Initial Investigation

Powers, et. al. [Ref. 4: p. 65] contend that an organization should establish standard procedures for dealing with user requests. They suggest that ideas for new or modified systems be examined and evaluated at a preliminary or exploratory level. The work performed is somewhat superficial: users must define their needs and come to an agreement on what is being requested.

The result is an understanding of the service request and what is to be done next. Possible alternatives include: (1) do nothing; (2) refer the request to a maintenance team; (3) refer the request to an information center (an entity within an organization specializing in user developed applications); or (4) move on to a more detailed systems analysis.

An initial evaluation should be a screening process to weed out those development requests that are not worthwhile and do so quickly to minimize the personnel expense involved in a study. Depending on the scope of the

request, an initial study may take anywhere from two days up to several months and may involve a single analyst or a team of analysts and users. [Ref. 22: p. 155]

When examining a request, the analyst(s) should gather background information on the situation and begin to assess the relative value of making the change. A cursory value analysis can be conducted by asking managers to place approximate figures on such items as lost revenues or increased operating costs because of deficiencies in existing systems. Requests initiated to comply with some statutory requirement should specify the mandated deadline and any penalties for late compliance.

Any intangible benefits flowing from an improved system should be defined in general terms. In some instances, a new system may affect other areas of the organization. When this possibility arises, the analyst should confer with the managers in the other areas to asses the impact of the proposed change on their operations. The acronym, IRACIS (Increase Revenue, Avoid Cost, Improve Service) has been used to summarize these basic objectives. [Ref. 22: pp. 155-156]

Besides monetary and intangible benefit considerations, the analyst and user must clearly understand and agree on the causing problem that was initially described in the request. Symptoms must be separated from the actual causes or a more costly redefinition of the problem may result in a later phase of the development.

Problem definition should begin with statements of the business objectives of the user area for which the systems request has been made, the responsibilities of the area, and the decisions that must be made by its managers. Ultimately, all systems modifications and improvements will have to be justified based on these objectives.

Logical systems objectives, the results the user expects to see, should be stated precisely in the user's business terms. Emphasis should be placed on the solution to the request not on physical requirements such as how the processing will occur. In other words, the investigation should concentrate on topics related to the need for preparing statements and reports and not whether it could be done on any particular computer or word processing system. [Ref. 4: pp. 73-75]

The existing system and procedures must be examined in order to understand how and to what extent they serve current operations. The major input sources and outputs for manual and computerized functions would also be reviewed.

A determination can now be made based on the characteristics of the existing system and the service requirements of the new request. The analyst would apply his knowledge and judgement to the question of whether the existing system can be modified to handle the new requirement or whether a new system will be needed. Furthermore, the systems analyst should consider several alternatives to the proposed solution, particularly when a detailed feasibility study is recommended.

Possible options may to suggest improvements to a currently manual operation without actually automating it or to provide partial sclutions as the alternatives. Gane and Sarson [Ref. 22: p. 167] have developed a simple "menu" to categorize the various levels of development effort and end products:

1. **The "hamburger" solution.** A low-budget, reasonably quickly implemented system which meets only the most pressing needs of the users' objectives, though hopefully adaptable to allow a more elaborate solution later

2. The "fried chicken" solution. A medium-budget, medium-time-scale system which achieves a majority of the users' objectives, but most likely not the most ambitious ones
3. The "chateaubriand steak" solution. A higher-budget, lengthy project which will achieve all of the users' objectives and have a major impact on the organization

Descriptions of features that should be incorporated in a new information system development is only part of the problem solving process. Financial, technical, and people-related constraints limit an organization's ability to implement desired system changes. Thus, no initial investigation would be complete without considering the factors that will influence successive development activities.

2. Feasibility Study

Any project may be considered feasible given that enough time and unconstrained resources are available. Reality is not so generous. Information systems development is more likely to be subject to a scarcity of resources and a tight delivery schedule. It is both necessary and wise to evaluate the feasibility of a project at the earliest possible time. Months or years of effort, thousands or millions of dollars, and professional embarrassment can be averted if an ill-conceived system is recognized early in the planning phase. [Ref. 23: p. 45] The feasibility areas that are of primary interest when performing an assessment include:

1. Economic or Financial Feasibility. An evaluation of development cost compared to the potential benefits, savings or income (i.e., "the bottom-line" analysis) derived from a proposed system.

2. **Operational Feasibility.** An evaluation of the impact on non-automated functions as a result of automating other functions
3. **Technical Feasibility.** A study of function, performance, and constraints (normally concerning the availability of existing software and hardware capable of supporting the system) that may affect the ability to achieve an acceptable system
4. **Schedule Feasibility.** A determination based on available resources and authorized expense levels that the project can be accomplished by a specific deadline.
5. **Legal Feasibility.** A determination of any infringement, violation, or liability that could result from development of the system
6. **Human Factors Feasibility.** An evaluation of anticipated personnel reaction (i.e., resistance to change) that could result from development of the system.
7. **Alternatives.** An evaluation of alternative approaches to the development of the system.

There are circumstances where economic justification is obvious, technical risk is low, few legal and personnel problems are anticipated, a flexible schedule is adopted and no reasonable alternative exists. More likely, one of the preceding conditions will introduce unacceptable risks and require management action. The success of the project depends on how extensively planners look at these feasibility considerations. A cynical, if not pessimistic, attitude should prevail.

The contents of the feasibility report should contain reliable, accurate assessments. Although the feasibility study may attempt to cover exhaustively all considerations, there are elements of risk in every new

development effort. Cash, et. al. [Ref. 24: pp. 313-319] point out that there are at least three important dimensions in a project that influence risk:

1. **Project size.** The larger the project in dollar expense, staffing levels, elapsed time, and number of organizational units affected by the project, the greater the risk. Multimillion-dollar projects obviously carry more risk than $50,000 projects and, in general, affect the organization more if the risk is realized. A related concern is the size of the project team's previous development efforts. The implicit risk is usually lower on a $1 million project for the team that is accustomed to working on developments in the $2 to $3 million range than on a $300,000 project for a development group that has never handled a project costing more than $50,000.

2. **Experience with the technology.** Because of the likelihood of unanticipated technical problems, project risk decreases as the technical expertise of the project team and IS organization increases. A project that has slight risk for a leading-edge, large systems development group may have a very high risk for a small, less technically proficient group. Risk can be reduced in the latter case through the purchase of outside skills for developments involving technology that is in general commercial use.

3. **Project structure.** When the outputs and input sources of an application are well-defined, understood and relatively fixed, the development project is classified as highly structured. These projects carry much less risk than projects that are subject to the developers' judgement and vulnerable

to change. With a low-structure project, the users may not decide what the outputs should be, or may change their minds often, halting progress.

Each project should be evaluated as to its relative risks in each of these dimensions. In addition to determining a relative risk for an individual project, an organization should develop an aggregate risk profile of the systems that are being developed concurrently. An organization loaded with high-risk projects, for example, suggests that they may be susceptible to operational disruptions when projects are not completed as planned.

3. Evaluate and Decide

The outcome of the feasibility/risk assessment study is reviewed by the appropriate level of management. If the decision to go ahead with a new development is made, the systems analysis process is repeated (or reiterated) in the analysis and general design phase.

Analysis and general design is a refinement of the activities performed during the initial investigation. As such, much of the preliminary analysis is reviewed and reevaluated. The objective is to complete the analysis and general design phase with a comprehensive and accurate user specification that will permit a smooth transition to follow-on development phases.

While the initial investigation concentrates on building an understanding of existing systems, of the need that has brought about a request for change, and of the potential solutions to identified problems; in analysis and general design, the goal is to produce specifications for a new system that will meet user needs and requirements. End products of the latter analysis phase include graphical models, flowcharts, and data flow diagrams which represent a physical and logical view of the system. These graphics

combined with qualitative descriptions and feasibility data are the user specifications which can then be converted to a physical design (the actual hardware, software, and data base used to implement the system). Thus, in computer information systems development, the goals of systems analysis are to start with an understanding of the organization and end with a formal specification of user requirements.

H. BSP VS. SYSTEMS ANALYSIS

Although the BSP and Systems Analysis methods have many activities in common, each approach offers management a distinct avenue to planning systems development. Selecting either of these approaches (or one of the alternatives presented in the following chapter), depends largely on the organization's structure, management style and experience with CIS development.

1. BSP Strengths

The BSP does, however, help to formulate a long-range IS plan and avoids the piecemeal approach to development. Other structured approaches usually concentrate on a single application or project. For organizations that are relatively new to computer-oriented systems or undertaking a massive change in computer technology, the BSP can be a low-risk alternative. The study's management viewpoint and inclusion of the majority of user groups can minimize interface problems and make redundant functions obvious. The study's results reflect the users' ideas of how their information needs can be best served. And the commitment required from top management to conduct the study can carry on throughout development making it less of an obstacle to get expense authorizations than

methods involvi a more bureaucra ic review/approval cycle. Its long-range sion allows an or nization to budget years in advance and o develop IS res rces at a rate consistent with business growth.

2. BSP Weaknesses

The BSP methodology relies on the knowledge and involvement of primarily non-DP managers. While this necessarily increases user participation, the study's results may not produce the most cost effective or efficient system. emoving key managers from their regular duties to conduct th study for 6 to 8 weeks ma be impractical for some organi ations. If significant changes occur within the organizatio al structure or operations are radically altered, the information architecture must be reworked. The BSP methodology acknowledges this possibility but does not elaborate on how management should incorporate major changes in their original architecture. One simply may not be able to stick another "black box" into the information architecture and tell the DP staff to start automating. It could happen that the information architecture won't fit one's organization at all. After Fort Ord re orted its successful results to Forces Command (FORSCOM), 47 other installations were directed to conduct BSP studies and many of them ended without producing worthwhile results. One of its most touted strengths is also its greatest weakness, namely, the users who have to interpret the study's procedures and derive meaningful results.

3. Systems Analysis Strengths

Using structured systems analysis forms a collective mind of general business practices provided by users and computer technology techniques provided by analysts. Users get a concrete idea of the pr posed system from logical data

flow diagrams developed from narrative and physical views of their systems. Presenting the system in terms of logical data flow early in the analysis reveals misunderstandings and contentious issues. While it may be weeks with the BSP study before team members can see what they have created through their fact gathering activities, the systems approach allows the analyst, sometimes after only a brief discussion with the requestor, to sketch a rough picture of the proposal. Even if this diagram is wrong, it is much cheaper to change a piece of paper than to back down out of a BSP in its fifth week. The interfaces between the new system and existing systems are shown clearly on the data flow diagram. With BSP the interfaces between existing and proposed systems are indistinguishable until broken out in a post-study development phase.

The use of the logical model of the system allows users and analysts to avoid duplication of effort. In other methods, including BSP, the user specification is passed to a design/programming group who effectively reanalyze it doing much of the work of data and logic definition again.

The structured systems analysis method is a more elegant fit to a single project or one with unique requirements. It offers both a top-down approach and the flexibility to tailor a system to fill a void in an existing information system.

4. Systems Analysis Weaknesses

The benefits of the systems analysis approach are not free. There are, of course, some costs and potential problems associated with it. Orientation of the users and training of the analysts is required. It may be perceived as "changing the rules" and, if so participants must be taught how to use the analytic methods and graphics to improve their systems. Users must learn the terminology and

become more involved in technical matters related to computer-based solutions. This can be a difficult, if not painful, transition step. The graphical abstractions are a necessary evil as well as the formality, level of effort, and degree of detail encountered with this approach. It is easy for users to become disenchanted with the many hours of research and analysis that seem to produce few tangible results.

Design/programming personnel may resent being relegated to mere "coders" because the user specification is sufficiently detailed to begin writing programs. Rarely would this be the case, there is a large amount of "thought work" left to do in the detailed design and implementation phases. Finally, not all users may be appropriately involved or the analyst misses an opportunity to improve other systems, and the users could discover that they have a technically excellent system that doesn't provide the information services they need.

I. SUMMARY

In this chapter, two analytical methods used to plan CIS developments were reviewed. The BSP method which produces an organization-wide sho - and long-term information systems plan; and Systems Analysis which produces a user specification normally associated with a single project. Relative advantages and disadvantages between the two approaches were presented.

There are a number of other development alternatives to both BSP and Systems Analysis but are limited in scope. These other development options along with project management issues will be discussed more fully in Chapter 5.

V. SYSTEMS DEVELOPMENT METHODS AND PROJECT MANAGEMENT

A. INTRODUCTION

The following sections in this chapter will explore several alternate methodologies for systems development. Each method represents a variation, or in some cases, a unique application of systems development techniques.

These techniques are not theoretical. All have been used successfully in actual practice. They are diverse because no single method is suitable for universal application. The choice of techniques offers management the flexibility to tailor their development efforts to varying system needs.

B. THE SYSTEMS DEVELOPMENT LIFE CYCLE

The systems development life cycle (SDLC) is recognized as one of the earliest attempts of get control over the costs and schedule of CIS projects. By the late 1960s most business organizations had evolved from their initial installation of equipment relying on input from punched cards to more modern devices utilizing magnetic tape inputs.

Businesses found themselves undertaking major computer system upgrades to remain competitive. Some companies were venturing into state-of-the-art data base technology. It was about this time when traditional development methods began to falter.

Data processing personnel, using traditional "bottom-up" approaches of designing individual applications and then applying them to subsystems and systems, were being overpowered by rising user demands and increasing technological challenges. The solution was to adopt a

concept that had worked in designing and building sophisticated hardware systems, such as aircraft, within tight cost and schedule constraints. [Ref. 19: pp. 112-113]

Powers et. al. [Ref. 4: pp. 38-40] emphasize that development is only a part of the SDLC process. In the total scope of CIS, there are several major stages:

1. **Recognition of need.** A bonafide need or problem must be identified before development begins.
2. **Systems development.** A process, or set of procedures, is followed to analyze needs and develop systems to meet them.
3. **Installation.** A system comes into use. The installation phase is the important transition from development to ongoing operation.
4. **Systems operation.** The system must be maintained and updated to meet changes in the organization which it serves.
5. **System obsolescence.** The system matures. The time comes when it is both desirable and economical to replace existing systems with new ones.

In order to cope with the specific requirements of each of these stages, the SDLC is organized into five distinct phases. The first stage, the investigation phase, has been discussed in-depth in Chapter 4. It is briefly reiterated here to illustrate its relationship to follow-on development activities.

1. Investigation Phase

The primary purpose, in this phase, is to determine whether a problem or need requires a full systems development effort or whether another alternative is more appropriate. If systems development seems appropriate, then

a feasibility study is conducted to determine the economic and technological impact of initiating a new development effort.

2. Analysis and General Design Phase

The existing system is studied in more depth and the concepts and designs are developed for the new system. Defining the logical structure and specifications of the applications functions and determining the software and hardware architecture begins. Half of the total time and effort involved in systems development may have been expended at the end of this phase. Therefore, a project plan, specifying the allocation of resources and authorization to perform certain work should be fully implemented.

3. Detailed Design and Implementation Phase

In this phase, hardware and software specifications are refined. Most of the computer-oriented work takes place during this phase. Programming plans are established and programs are written and tested. Training materials and user procedures are prepared.

A trial system undergoes testing by select users that is extensive enough to result in either acceptance or specifications for further modification. If the system is accepted by the users, the steering committee (when one exists) is asked for approval to proceed with the installation phase.

4. Installation Phase

The chief purpose of the installation phase is to make the transition from existing procedures to new ones. Remaining users are trained and the old system is phased out. The impact of change is felt fully by the organization

and its people. Human discomfort and resistance to change can be extensive and serious.

During the installation phase, an investment in end user training may provide high yields by enhancing the value of the new system. Special demonstrations, briefings and continued consultations to help users understand the full potential of their system may be required. However, no amount of encouragement will overcome inherent deficiencies in the applications. Results speak for themselves, and user acceptance is only a partial measure of success. More definitive measures are evaluated in the review phase.

5. Review Phase

The review phase in the SDLC process is dedicated to looking back at the experiences and lessons learned during the first four phases. Powers et. al. [Ref. 4: p. 46] suggest two reviews should be made for each project. The first takes place shortly after the system has been implemented while the project team is still together. The team members should share the memories of successes and failures during the systems development effort. The main purpose is to help the organization improve the systems development skills it will carry to future projects.

The second post-implementation review takes place approximately six months after the first. The intent is to measure the results of the new system and compare them with the projections of system performance, in terms of benefits and savings, at the outset of the project.

C. VARIATIONS TO SDLC

How much time to spend on a particular phase may vary greatly from project to project. The key point is understanding the objectives of each phase and the

deliverables that users/analysts must produce. Tommela [Ref. 2: pp. 112-118] describes three variations--serial, overlapping, and functional approach--that have been used in SDLC management. Figure 5.1 illustrates the relationships between the three approaches.

1. The Serial Approach

With the serial approach, each SDLC phase is completed before the next begins. The applications are usually simple and straight-forward. The complexity and functions are easily grasped by the developer and partitioning the workload is an uncomplicated matter.

This approach, therefore, is best suited to projects of short duration (less than six months) and with limited staffing (approximately three people).

2. The Overlapping Approach

The overlapping SDLC approach may be used when an earlier delivery of small systems is desired or for projects of medium duration (six to twelve months) and staffing of approximately eight people.

In the overlapping approach, some phases begin before the preceding phase is finished. The applications are usually more complex and the subdivision of tasks is more difficult because of the interrelationships of application functions.

3. The Functional Approach

The third variation of the SDLC is the functional approach. It incorporates the same five phases as the serial and overlapping methods, but, the deployment of the phases differs significantly.

Using the functional approach, an application is analyzed hierarchically in terms of its discrete functions.

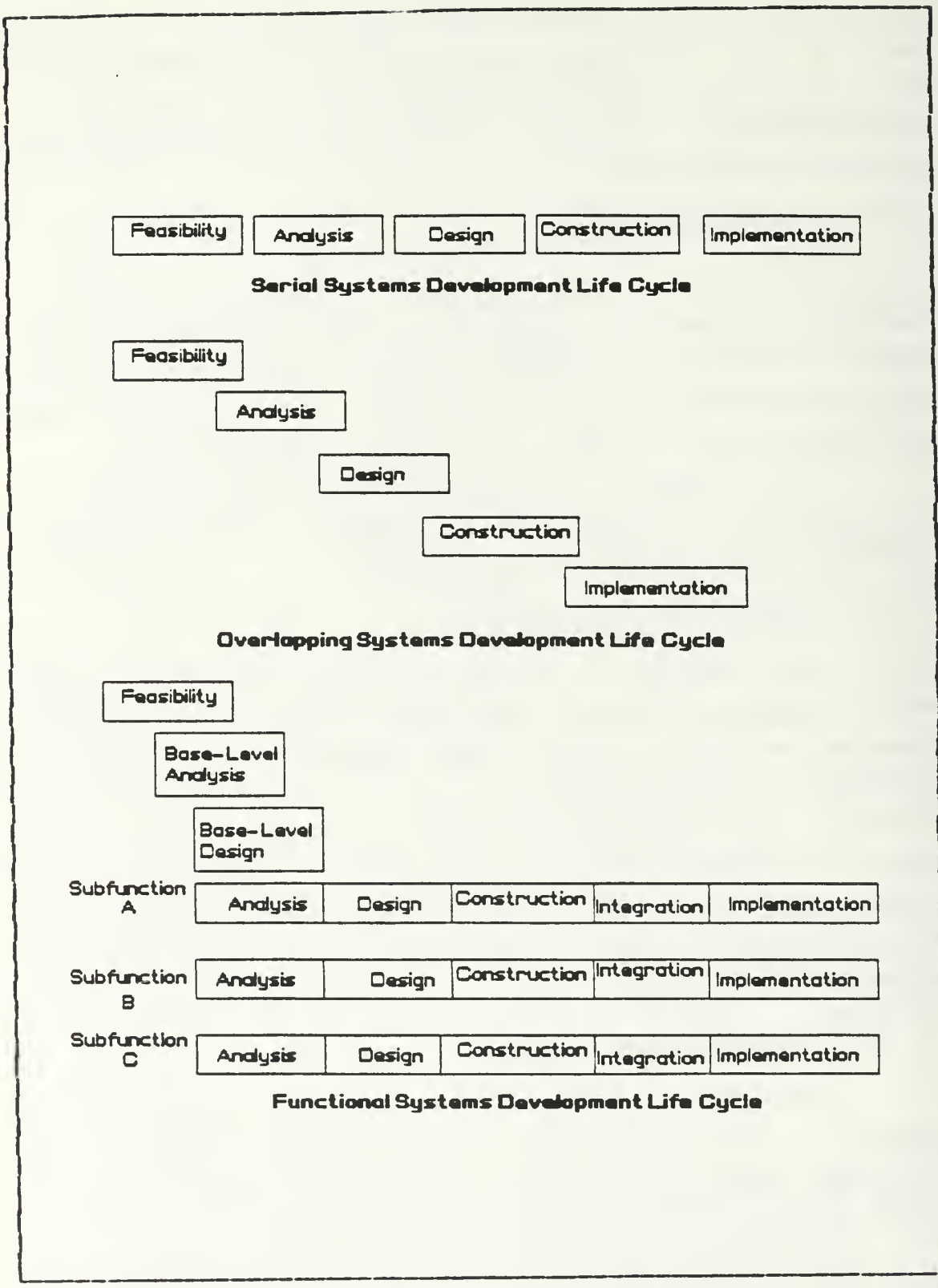

Figure 5.1 Comparison of SDLC Approaches

This results in the application being segmented into levels of subfunctions. Figure 5.2 illustrates a part of the functional hierarchy of a materials system.

Level 0 is the application being developed. Level 1 represents the major subfunctions of the application. Levels 2, 3 and below are the exploded components of their immediate, higher subfunction. The number of subordinate levels depends on the complexity of the subfunction. The use of the functional SDLC approach, coupled with structured techniques, permits each subfunction to be developed and implemented independently from and concurrently with other subfunctions. Thus, each subfunction can follow its own development life cycle.

After the first subfunction is implemented, the succeeding subfunctions pass through an additional development phase known as integration. Integration involves assembling the components into subsystems and ultimately into the overall system while ensuring that proper interfaces exist between components. The system then evolves as each subfunction is integrated with its predecessors.

D. SDLC LIMITATIONS

The SDLC approach has some notable limitations. It tends to be less responsive to changing user requirements than other methods. Users are expected to state their requirements clearly by the end of the analysis phase. Often, these user specifications require modifications that aren't discovered until the detailed design and implementation phase is well underway. By "revisiting" the analysis phase to make these changes, the development effort experiences higher costs and longer delays than anticipated. Tommela [Ref. 2: p. 114] discusses other problems with the SDLC method such as:

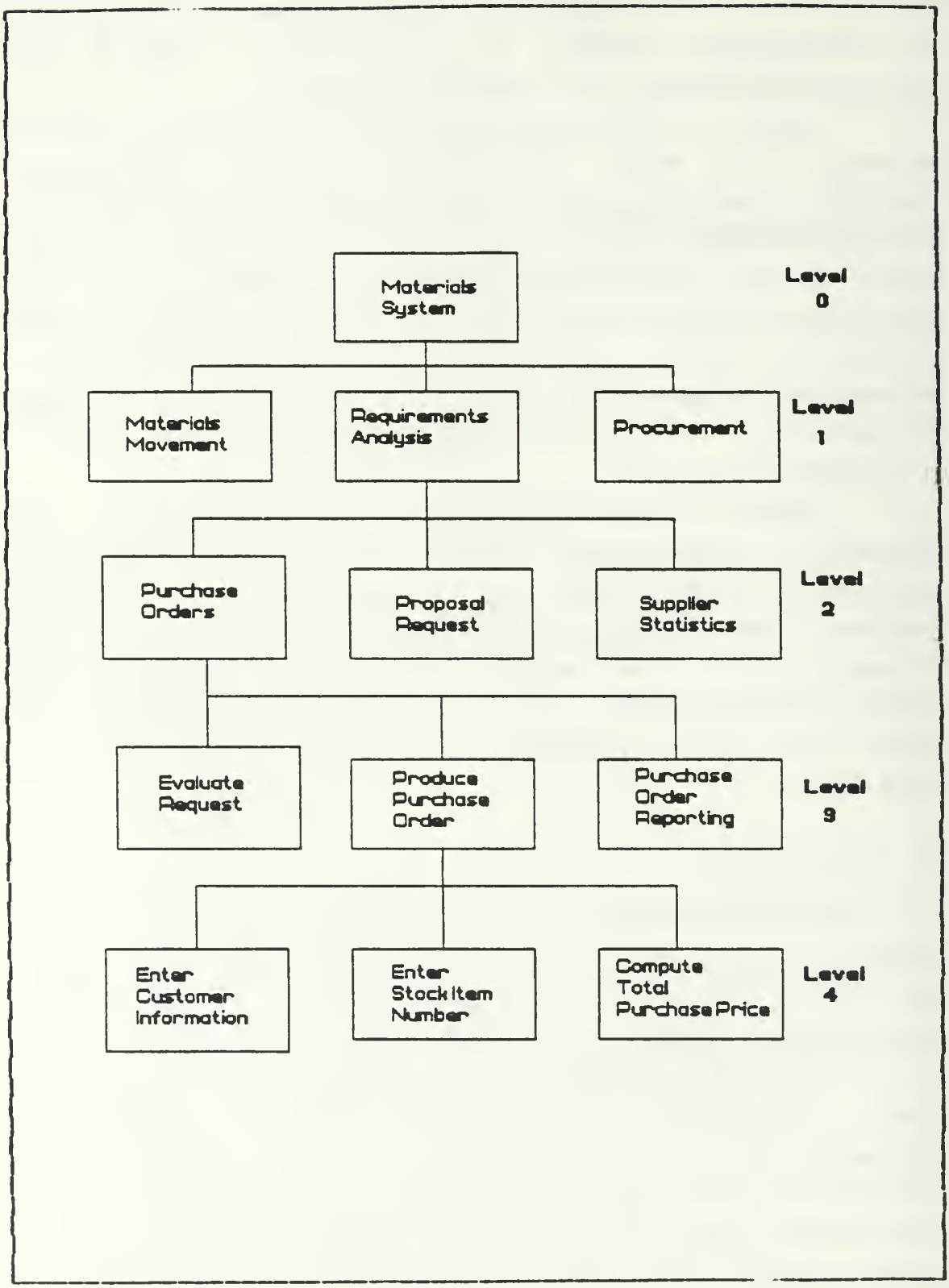

Figure 5.2 A Functional Hierarchy

1. Forcing the users to make premature decisions about a system they "can't see"
2. System designers pressuring the users to sign off or freeze requirements
3. An overwhelming number of functions to isolate and analyze for large, complex applications
4. The "Big Bang" implementation--stopping the old system one day and starting operations with the new system the next
5. The inevitable swell of the backlog of problem reports and user-requested enhancements.

While the functional approach does alleviate some of the inflexibility of the other traditional SDLC methods, user requested alterations are a normal part of the development process. Implementing them, in the SDLC environment, is the usual cause for cost and schedule overruns.

It can take years to implement some large scale systems using SDLC methods. These long-term developments are vulnerable to high personnel turnover, cost overruns and intense user dissatisfaction. Fortunately, for managers, more progressive alternatives are available.

E. HEURISTIC SYSTEMS DEVELOPMENT AND PROTOTYPING

1. Heuristic Systems Development

The heuristic approach to systems development refers to a methodology which allows the systems analyst to define user requirements by trial and error while designing the output system. It is sometimes called the "iterative" approach. Wetherbe [Ref. 20: pp.162-163] describes the activities of the heuristic development as follows:

1. During the analysis phase, develop a broad understanding of the data currently used to support decision making and operations.

2. Obtain samples of machine-readable or manual data and load them into a data base as simple sequential files.
3. Determine fields to use as indexes and establish any obvious relationships using the technology provided by a data base management system (DBMS).
4. Using a query language, develop screen and report formats based on information currently required by users. Devise any additional screen formats that could be useful.
5. Train the users in the operation of the system and allow sufficient time for users to interact with most of its features. This experience encourages the users to more fully envision and articulate their information requirements.
6. With the information gathered in Step 5, revise the system by:
   a. Adding new fields
   b. Creating new data relationships
   c. Modifying screen formats
   d. Eliminating seldom used indexes to improve performance
   e. Coding frequently used queries into a higher performance language such as COBOL to increase the response rate
7. Repeat (iterate) steps 5 and 6 until the system is relatively stable.
8. Design an input system to provide edit and update capabilities for the data structure and the output system. Then proceed with the remainder of the development cycle.

Developing the output system before designing and developing the input system is a logical sequence. Developing an input system is usually a major effort. When

the output system developed accurately fits user requirements, the input system is easier to define and is less susceptible to change. Another approach, which permits the users to evaluate outputs before the system is fully implemented, involves prototyping. A prototype is a smaller scale version of the target computer system.

2. Prototyping

Prototyping, like heuristic development, is a strategy that allows user requirements and systems design to evolve together. The basic reason for selecting the prototype approach is that it is easier to make changes to a system when it is not fully installed throughout the organization. Many minor defects can be identified and corrected for the following day's testing. Wetherbe [Ref. 20: p. 163] outlines four major steps to prototyping:

1. Identify the users' basic information and operating requirements.
2. Using a small representative data base, develop a working prototype which performs only the most important, identified functions.
3. Demonstrate the prototype and allow a test group of users to interact with it. Development team members should sit alongside users operating the system to observe their actions and to elicit change recommendations.
4. Incorporate the user requested changes in the next version. After the next prototype is implemented, repeat steps 3 and 4 until the system fully achieves the requirements of the users.

The duration of the prototype depends on many factors, including application complexity, number of changes identified, and hardware limitations. The most important criteria when using the prototype approach is to make all

needed changes before the system is expanded to include all users. Changes can range from the reformatting of data on a screen to the complete redevelopment of a function. It seldom makes sense to provide the system to all users when it is evident that the system cannot meet performance specifications.

Prototyping offers an excellent opportunity to measure the system's impact on network and computer resources. This is often overlooked and results in users who are dissatisfied because response time at the terminal is twice as long as originally planned. The prototype should be conducted long enough to check network management procedures for telecommunications failures, computer failures, and requests for vendor assistance. A prototype offers the two best results that developers can expect with a new project: an exceptional opportunity to implement an system free of errors tailored to user needs and end users who are pleased with the development end products that they helped to design.

F. BENEFITS OF HEURISTIC AND PROTOTYPING APPROACHES

The benefits derived from the heuristic and prototyping approaches include relatively shorter development times, more accurate determination of user requirements, greater user participation and support, rapid response to user requested changes and a less threatening process of design specification and implementation for both the systems architects and end users. Integrating the heuristic and prototyping approaches with an organization's formal SDLC methodology may be done following the guidelines in Table 1 [Ref. 20: pp. 165-166].

TABLE 1

Development Method Variations

SDLC	Heuristic	Prototyping
Problem/ opportunity Definition	1. Develop understanding of data to support decision making and operations.	
Analysis		
	2. Collect samples of data and load into data base.	1. Identify basic operating and information requirements.
Design		
	3. Establish initial indexes and interfile relationships.	
Technology/ Personnel Requirements		
	4. Using DBMS query, develop screen formats/reports.	2. Develop working prototype.
		3. Demonstrate prototype.
Develop, Test, Validate	5. Train users to work with output.	4. Refine and make new versions of prototype and repeat step 3 onward until system is complete.
	6. Modify system based on what users discover in step 5.	
	7. Repeat 5 and 6 until system is relatively stable.	
Implement		
Evaluation		

For an organization to incorporate the heuristic and prototyping methods into its SDLC process, the key advanced technologies of on-line interaction, DBMS, and query-based languages must be in place. The development team must be educated in the process and a few progressive systems developers should use the techniques on several small projects. After successfully completing these small projects, larger ones can be addressed and more staff encouraged to use these advanced methods. [Ref. 20: p. 167]

G. PROJECT MANAGEMENT

While the CIS planning process focuses on a multi-year view of matching technologies and systems to the organization's evolving needs, project management concentrates on formulating a system which guides an individual project's life cycle. Many of these methods and tools have been described in the previous chapters. Much of the literature and conventional wisdom suggests that there is a single correct way to manage projects. The notion is that managers should apply uniformly the tools, methods and organizational structure to each development effort.

While there may be a generalized set of methodologies, the contribution each device makes to planning and controlling a project varies widely according to the project's characteristics. In short, there is no universally correct way to manage all projects. Cash, et. al. [Ref. 24: p. 320] refer to four principal types of project management "tools" that should be balanced according to the type of development being undertaken. Table 2 [Ref. 24: p. 321] gives some examples of the tools in each category currently being used by various organizations.

TABLE 2

Project Management Tools

External Integration Tools	Internal Integration Tools
Selection of user as project manager	Selection of experienced DP professional to lead team
Creation of user steering committee	Selection of manager to lead team
Frequent and in-depth meetings of this committee	Frequent team meetings
User-managed change control process	Regular preparation and distribution of minutes within team on key design evolution decision
Frequent and detailed distribution of project team minutes to key users	
Selection of users as team members	Regular technical status reviews
Formal user specification approval process	Managed low turnover of team members
Progress reports prepared for corporate steering committee	Selection of high percentage of team members with significant previous work relationships
Users responsible for education and installation of system	Participation of team members in setting goals and deadlines
Users manage decision on key action dates	Outside technical assistance

TABLE 2

Project Management Tools (cont'd.)

Formal Planning Tools	Formal Control Tools
PERT, critical path, etc. networking	Periodic formal status reports
Milestone phases selection	Change control disciplines
Systems specification standards	Regular milest ntation meetings
ibility study specifications	Deviations from plan
Project approval processes	
Project postaudit procedures	

Each of the four categories serves a special purpose in the development environment. Managers must match the proper tools with the type of computer project that is undertaken and the people who will perform the development tasks. Each category of tools can be briefly summarized as:

1. External integration tools include the organizational and other communications devices that link the project team's work to users at both the managerial and lower levels.
2. Internal integration tools are those devices that ensure the team operates as an integrated unit.
3. Formal planning tools help to structure the sequence of tasks in advance and to estimate the time, money and technical resources the team will need to achieve the project's objectives.
4. Formal control mechanisms are those devices that help managers evaluate progress and spot potential discrepancies so that corrective action can be taken.

Structure and technology are two primary factors in projects that influence how the management methods and tools should be applied. The term structure implies the arrangement and relationship to interdependent parts in a computer information system. Technology, related to CIS projects, involves an understanding of the technical methods for achieving the solution. Cash, et. al. [Ref. 24: pp. 321-326] suggest that managers categorize projects by their relative levels of structure and technology and evaluate the risks accordingly.

1. High Structure-Low Technology

High structure-low technology projects present familiar technical problems, have minimal risk and are the easiest to manage. They are also the least common. Outputs are very well defined by the nature of the task and the

users are less inclined to change their minds about expected end products.

Extensive administrative procedures to get a diverse group of users to agree on specifications are not necessary. Inclusion of analysts in user departments, heavy representation of users on the design team, and formal approval of design specifications are cumbersome for this type of project. Training users, however, to operate the new systems remains an important integrating device.

The technology in these projects is familiar to participants. A high percentage of persons having only average technical backgrounds and experience can be involved. The team leader does not need strong computer systems skills which makes this type of project suitable for junior managers to run and gain some experience.

Project life cycle planning concepts with their focus on defining tasks and budgeting resources against them, force the team to develop a thorough and detailed plan. Such projects are likely to meet mandatory milestone dates and keep within the target budget.

2. High Structure-High Technology

High structure-high technology projects are vastly more complex than high structure-low technology developments. They involve significant modifications to the procedures outlined in the project management methodologies. Conversion of systems from one computer manufacturer to another is a typical example of a project that is a high structure-high technology development requiring tight controls.

Outputs, as in the first type, are well defined and their susceptibility to change is low. However, liaison with user groups should be more intense to ensure coordination on any input-output changes to the

specification and to deal with any systems restructuring that must follow shortcomings in the project's technology. This type of project normally encounters problems because the technical system developed is inadequate to fulfill the users objectives.

The team leader must possess the administrative skills (not necessarily data processing knowledge) required by any project of technical complexity. The leader must be effective in communicating with technicians. His ability to establish and maintain teamwork through meetings, document all key decisions, and chair subproject conferences is critical to the project's success.

Project life cycle planning methods, such as PERT (program evaluation and review technique) and critical path method (CPM) are used extensively but their predictive value is much more limited than for projects in the first category. The team may not understand key elements of the advanced technologies being used and seemingly minor program defects can become major financial drains.

Technical leadership and high internal integration devices are keys to this type of project. Formal planning and control tools tend to provide more subjective than concrete projections. The danger is that project managers and decision makers may believe they have precise planning and close control when in fact they may have neither.

3. Low Structure-Low Technology

Low structure-low technology projects pose low technical risks but may fail because of inadequate direction. Since there may be numerous, well-known technical alternatives that could be applied to the problem solution, the difficult management task is obtaining user commitment to a specific design.

The specification and design of user requirements must be rigorously controlled or the project manager may be bombarded with change requests. And the importance of tough, pragmatic leadership increases once the design is final. Some type of formal change control process may be necessary to limit modifications to only those of strategic significance.

Formal planning tools are useful in structuring tasks and helping to remove uncertainties. The system delivery date will be firm if the specifications remain relatively unchanged. Formal control devices are normally effective for tracking progress and identifying schedule slippages or advances. Because technology problems are low, a staff with varying degrees of technical backgrounds should be adequate. The key to success is close, aggressive management, but the leadership must come from the user rather than the technical side.

4. Low Structure-High Technology

Low structure-high technology projects are complex and carry high risk. Team leaders need sound technical knowledge and experience, and the ability to communicate well with users. Total commitment on the part of users to a particular set of design specifications is vital, and again they need to agree on one, out of many, technical alternatives. The greatest risks with these projects is that the user perspective may turn out to be infeasible in the selected hardware/software solution for the system. Technical complexity makes strong technical leadership and internal project control essential. This kind of development effort requires the most experienced project managers and will need wholehearted support from the users. The project manager usually must decide whether the effort can be divided into a series of much smaller projects or may use less innovative technology.

Formal planning and control tools are useful but contribute little to reducing uncertainty in the early planning stages. These tools do allow the project manager to structure the sequence of tasks but with this type of project new tasks crop up with regularity. Tasks that seem simple and small may become complex and protracted. Time, cost and resulting system performance are extremely difficult to predict simultaneously. If cost and time considerations give way to technical performance, the outcome may be unacceptable to the users who are paying for the system.

Deciding which approach to take in putting together a project can mean the difference between success and failure. Managers, using the preceding guidelines, can shape their strategy to fit the needs of individual developments.

5. Project Management Software Tools

Project management software can help reduce the clerical support and time spent planning and controlling projects. Any manager who spends substantial effort overseeing computer systems developments can benefit from using one of these products. These software packages are not limited to computer-oriented projects. They can be used to automate many of the widely practiced management methods whether the project involves construction of a building or a mass-transit system. Many of these project management software products are available in microcomputer versions making them more portable and appealing to a larger group of users.

A project management tool will not substitute for good management practices or overcome unrealistic expectations, inadequate resources or poor workmanship. They can be used to help specify what will happen, who will

do it, when it will get done and how much it will cost. In short, these products cannot tell managers what to put into their plan, but they can help to manage whatever is put in. These systems will help managers to look at their plan, measure the allocated resources against the plan, keep track of progress and bring the project to its fruition. [Ref. 25: pp. 104-106]

Project management software can be invaluable in the initial planning stages of a project. Limitations, inconsistencies and activity overlap can be uncovered quickly. Individuals can be assigned tasks in the correct priority and sequence minimizing the tendency to do the easiest job first not necessarily the most pressing job. Scheduling personnel, facilities and other resources is simpler than manual methods.

Some products have a "what if" analysis capability. This feature is particularly useful on projects that involve high uncertainty in technical or human issues. The ability to forecast proposed changes, analyze feasibility and add necessary resources helps managers control the creeping scope of projects. By using the "what if" capability, managers cannot only determine how many and how long but how best to allocate vailable resources. A project manager can then tailor his development effort with the most acceptable combination of time and resources.

Acquiring a project management system can be as formidable a task as buying any other type of software product. A package must be fully functional but not a project in itself to learn and operate. If it's too hard to understand or forces an overly bureaucratic and cumbersome approach, it will not be used. The managers who normally guide development work should be the primary input when selecting these systems.

Once the system has been purchased, reports generated and assignments and deadlines are clear and tolerable to all, project managers can use the package to continually remind everyone what must be done. The key human issue in project management is the insistence of quality. Some personnel will resent the use of such systems because they readily illuminate poor management practices and inefficiency. Although these products were not designed to identify poor performers, per se, they can help to weed out underachievers. Taking a poor performer off the development team can often be more productive than adding a good one. On the subject of people and quality, DeMarco [Ref. 25: p. 197] relates this story from his days as a design instructor:

> "I was presenting a seminar to a project team on the West Coast. There were about twenty people in the class, including two hardware types. These two had had only a single programming experience between them--a piece of software they had built together some years before. The program was still alive and well, and had earned them considerable renown; throughout its years of use, no one had ever found a bug in it. I asked one of them how he explained this phenomenal success, and apparently bug-free delivery on first try. 'Well,' he said, 'we didn't know bugs were allowed.'"

If an organization is fortunate enough to have such people as these two hardware engineers, they may have the best system for keeping a project out of trouble. If not, a good project management package can help managers keep the quality and timing of development efforts in check.

H. SUMMARY

Planning to do a project is one thing but doing it correctly is another. In this chapter, common development methodologies, advanced software techniques, and human and technical issues in project management were investigated.

There is much more to consider, of course, but the problems discussed represent what managers can expect with CIS developments.

The inherent complexity of applications development requires subdividing the target system into manageable components. The preferred arrangement is to follow the functional SDLC for large scale, highly complex projects and to use heuristic and prototyping methods to evaluate subsystems. The latter two methods are also useful for developing small computer systems or individual applications to fill voids in an existing system. The heuristic and prototyping methods, however, require that certain advanced technologies be in place before they can be used. Sophisticated software packages for manipulating data are a key part of these technologies. The use and importance of applications development software are addressed in the following chapter.

VI. USER-ORIENTED DEVELOPMENT TOOLS

A. INTRODUCTION

The heuristic and prototyping development methods require software tools such as DBMS packages, fourth generation (4GL) and query languages. Besides development, there are many other uses for these packages. Application developments, however, represent large outlays of money and personnel effort. It is in this area that sophisticated software packages offer the highest potential gains in productivity. Several thousands of these products are currently in use and trends suggest many thousands more will be purchased in the next few years.

For a business to get the most out of a fourth generation language or other software development product, decision-makers must understand what the technology offers and they must have a clear understanding of their organizational needs. This chapter investigates the capabilities of many of the software development tools which are helpful in the construction and maintenance of user requested applications. These software tools coupled with the development methods in the preceding chapters create an environment where users can assume some of the DP workload and contribute to the overall productivity of their organizations.

B. A MANAGEMENT DILEMMA

Selecting a a fourth generation language, query or DBMS package is difficult because it may make the organization dependent on these tools and on the systems put in place through their use. Packages may be purchased in response to

a specific user need or an integrated product set can be obtained which addresses a wide range of needs, but, which may also require a much greater commitment to the support and usage of that product. Managers must also be aware of the possibility that the software vendor who provides technical support for these packages may not survive in the highly competitive computer market place. Steps can be taken, e.g., placing the software object code in escrow, to protect yourself but the best action is to do the necessary research to find a reliable vendor. [Ref. 27: pp. 27-28]

1. No Standard Definition for 4GL

One of the chief problems that can be encountered when reviewing fourth generation, DBMS or query language features is the lack of any standard definition for these packages. They are generally lumped into one category under the heading fourth generation languages (4GL). Snyders [Ref. 28: pp.28-30] confirmed this dilemma when she received the following responses from industry experts:

"There's no formal definition of a 4GL."

Dave Litwack
Vice President
Cullinet Software

"The only characteristic that 4GLs have in common is that they are not COBOL."

Stephen Gerrard
Product Marketing Director
Applied Data Research

"A fourth generation language is basically any computer language that is nonprocedural."

Richard Cobb
President
Mathematica Products Group

"The cardinal hallmark is that with a 4th generation language, a user specifies what to do, not how to do it."

David Wszolek
Director of Marketing
Information Builders

"4th GL is a language that dramatically increases the productivity over another language such as COBOL or Fortran."

Chuck Riegel
Senior Marketing Representative
Software AG of North America

Although there appears to be no standard definition, most fourth generation languages fall into distinctly different categories. These categories represent various features and user expertise levels that the packages are directed toward. Fourth generation languages can be classified as: those developed by data base management system (DBMS) vendors and non-DBMS vendors; formal versus informal languages; procedural versus nonprocedural; batch versus on-line; and professional versus nonprofessional users.

The suppliers of 4GL are divided into two major groups. DBMS vendors such a Applied Data Research, Cullinet and Software AG offer products that are the primary DBMS in an organization. Other suppliers include Information Builders Inc. (FOCUS), Mathematica Products Group (RAMIS II) and Dunn & Bradstreet Computing Service (NOMAD 2) who develop fourth generation languages that support different data base systems such a IMS (the "first" commercial DBMS) from IBM.

Most of the key distinguishing characteristics of software development tools can be determined by how they are used and who uses them. Santarelli [Ref. 29: p. 22] has further subdivided the DBMS and fourth generation language product by category to emphasize user features. Examples of these products and their corresponding capabilities are provided below:

1. Query and reporting tools such as ASI Inquiry from Applications Software and Mark V from Infomatics.
2. Fourth generation programming languages that offer increased productivity to COBOL programmers such a

ADS Online from Cullinet, Mantis from Cincom and Ideal from Applied Data Research.

3. Information Center products targeted at non-programmers such as NOMAD 2, FOCUS and RAMIS II.
4. COBOL program generators for experienced programmers such as TELON from Christiansen Systems and IP-3 from Computing Productivity.
5. Decision Support Systems (DSS) designed for analyzing and extracting data which include System W from Comshare and Express from MDS.

These packages offer significant benefits in terms of increased productivity and end user solutions in applications development. The type of vendor and features that an organization should choose in selecting a fourth generation language will follow, in part, the type of enterprise they pursue, in-house programmer expertise levels, and the information processing workload that must be handled by the 4GL package. In the following sections the fourth generation languages and DBMS products that will be discussed largely refer to mainframe and minicomputer systems. Appendix C contains a representative sample of the various products currently available including several microcomputer versions.

2. General Characteristics of 4GL

The evolution of fourth generation languages began with the transition from machine language (first generation binary digits or "bits") to assembly language (second generation alphanumeric characters). This stage brought approximately a seven-to-one advantage in productivity and the ability to write and develop programs. Third generation higher level languages such as Fortran, Basic, PL/I and Cobol were developed bringing a seven-to-one improvement in productivity over assembly languages. These languages

became the building blocks of today's 4GL software tools. [Ref. 30: p. 24]

Fourth generation languages are sometimes referred to as non-procedural. Actually they are just less procedural than their predecessors. The term procedural means that the user (or programmer) must specify not only what he wants to accomplish but must describe in detail (via a program) to the computer the sequence in which to execute the required steps. Open the file, read, record, create a counter, add one to the counter are examples of programming steps that would be specified with a procedural language. Fourth generation languages eliminate these details earning the non-procedural classification.

Non-procedural 4GLs use English-like or other natural language commands to allow the user to manipulate data. Natural language systems, either provided with the vendors 4GL product or purchased separately to interface with another vendor's products, convert human language to computer useable forms. The commands, therefore, are easy to learn, use, and support.

Many fourth generation languages can print their own documentation, simplifying application updates or changes. They are easy to transport from computer to computer, and applications developed with them move between these computers without change. Fourth generation languages use a virtual memory-based design to reduce memory requirements by permiting blocks of data to be exchanged in appropriate portions of the program as they are needed. [Ref. 30: p. 24]

Fourth generation applications can accommodate small specific business applications or can be used to customize large, existing or off-the-shelf software programs. Since most of the documentation is contained in the 4GL applications, the loss or impending loss of key programming

personnel is less threatening to ongoing operations. Substantial reductions in time and expense for user training can be realized with 4GLs.

3. Fourth Generation Languages Are User Constrained

Not everyone can use a fourth generation language. User friendliness can only be specified by what a particular user finds friendly and what the user wants. Someone who has never used a computer terminal keyboard cannot perform easy fourth generation language tasks.

Realistically, a user must reach three levels of computer sophistication. The first level is computer awareness. The second level is achieving proficiency in the use of a presentation language, menu-driven screens, or natural language commands for performing simple inquiry tasks. Next, the user must become familiar with more advanced commands and applications to perform more complex tasks (manipulating files, sorting records, compiling reports, etc.).

When the user reaches the third level of computer proficiency he will be relatively expert and be able to use the highest level category of fourth generation software. At this level, the user can define procedural processes, develop applications and be comfortable with working throughout the range of 4GL capabilities. Users at this third level can sometimes develop projects as big as those traditionally handled by the organizations' DP department. [Ref. 29: pp. 27]

One potential drawback to 4GL is that users may solve problems from their perspective not from an organization-wide perspective. When it comes to large projects, encompassing the entire business, the task will still have to be centrally managed by an information systems development team and not through a collection of end user

activities. The management information systems (MIS) departments or other DP service groups within an organization will have to tailor their 4GL support to meet varying degrees of user proficiency. For some user groups, the MIS staff may control all phases of software use providing basic data manipulation capabilities and locked applications (unchangeable by the users). Other user groups can be given a capability to modify existing applications utilizing a more advanced set of manipulation commands. A third user group may participate fully in application development. This is essential for those specialized tasks that functional area users know best. Users with little or no programming experience can create applications using the full range of 4GL features. The more adept they become, the more imaginative their application of the language becomes. [Ref. 30: p. 24]

Another user related problem is the accessibility to appropriate data. The features of 4GLs are only useful if the appropriate data is accessible for inquiry and presentation in a reasonable way. By and large, information within many organizations is not positioned for easy accessibility. MIS staffers will have to work to overcome this problem by setting up information centers, dedicating special computer systems, devising new data bases, and periodically replicating information from different sources to customize data bases for a large end user community. [Ref. 27: p. 27]

4. Purchasing a Fourth Generation Language Package

An organization can take two basic approaches when purchasing a fourth generation language: acquiring a specific tool for a specific need, or purchasing an integrated product set which covers a wide range of needs. The difference in cost between these approaches can range from tens of thousands to hundreds of thousands of dollars.

Another major consideration is selecting a vendor that can provide technical support and be around when you need them. Software does not wear out but "bugs", program defects, can appear creating a nasty problem for your organization. Another reason for careful vendor selection concerns technical support. Organizations should own a product where periodic improvements are provided by the vendor. While there may not be a sure method of picking a vendor to suit all fourth generation needs, some software companies are emerging as "mega-software vendors." These companies are developing product lines that run the gamut from microcomputers to mainframes. Vendors such as Cullinet, Applied Data Research and Mathematica Products Group are developing a total business system of integrated products and application tools to meet varying informational needs within an organization. [Ref. 30: p. 24]

5. Future Fourth Generation Language Trends

Fourth generation languages have helped to take the power of computing to the end user. Because of these tools, and our increasingly computer-literate society, far more people will be able to share the applications development work and improve an organization's productivity. This is of particular significance to governmental agencies. The U.S. Office of Management and Budget's (OMB) "Management for Fiscal Year 1986" report stated that steps must be taken to "recapture the government's position as leader in the efficient and productive use of information technology." [Ref. 31: p. 16]

Software costs today amount to 60% of federal computer expenditures, compared with 20% in 1965. Additionally, the federal government continues to custom-develop 90% of its software and the transition to modern, efficient hardware is inhibited by large volumes of

custom code that require conversion. Beginning in fiscal year 1986, government agencies will be asked to reduce their software costs by 25 percent and their software staffs by 5,000 full-time positions over the next three years. One primary means to reduce software costs will be through the use of fourth generation languages for applications development. [Ref. 31: p. 16]

There are estimates that soon, more than 50% of all programming activity will be done by end users. A few years ago, that percentage was essentially zero. When you can develop end user applications utilizing a 4GL product in one quarter to one tenth of the time it takes in COBOL, and at one tenth the cost it takes to maintain a COBOL solution, these trends will likely increase. [Ref. 27: p. 28]

C. DATABASE MANAGEMENT SOFTWARE

Management of an organization's investment in computer information systems is increasingly focused on methods to improve control, consistency and coordination in the development and support of applications for end users. Often these methods are based on database management systems (DBMS) technologies. Another key tool to assist management in controlling its data resource is known as a data dictionary system (DDS).

DBMS and DDS have introduced more than just an innovative means to transform data into information; they have brought revolutionary changes to an organization's information systems structure and operations. More and more, businesses are modifying their traditional DP organizations to meet the broadening functions of information management. Positions such as a database administrator (DBA) and data administrator (DA) are being established in recognition of the specialized needs of DBMS and DDS technologies.

These two tools in particular are literally changing the way managers regard data and information. Data/information is increasingly seen as a resource that requires administrative procedures and controls just as money, personnel and facilities have had all along.

D. DATABASE TECHNOLOGY

In the early 1970s, E.F. Codd and C.J. Date, published a mathematical approach to defining and manipulating the concept "data". Their work revolutionized the way we would come to design, organize and access databases. Codd and Date were primarily pursuing an academic exercise. They were out to rename the vague empirical terms then in use in favor of more rigorous mathematical definitions for data itself and for the operations that can be performed on data. These two men wanted to lay a foundation for data analysis; they had no intention of developing software to implement their hypothetical programming language. Yet, when their work went public, many DP organizations wanted to buy one. And software vendors, more or less, produced their versions of these concepts calling them database management systems (DBMS). [Ref. 32: pp. 118-120]

1. Database Concepts

"A database is a collection of data that are shared and used for multiple purposes," according to Martin [Ref. 33: p. 4]. Database technology reduces the artificiality imposed by separate files for separate applications. It allows an organization's data to be processed as an integrated whole and permits users to access data more naturally. The predecessors of database systems were file processing systems. With file processing systems, each data file is considered to exist independently. One of

the major limitations of file processing is that there is no guarantee that the files are compatible. One file may be written in COBOL binary format while another is coded in an incompatible PL/I record format. When this is true, one file must be converted to the format of the other, and an extraction program written, tested and run. This process can represent an unacceptable delay to users. End users may decide that responses to new requirements or ad hoc requests are sc long in coming that they are not worthwhile. [Ref. 33: pp. 2-3]

With a database system, files are integrated into a database. These files are logically "tied together" by relationships between records or data items contained in the files (actually the data files can be located on physically dispersed storage devices such as magnetic tapes, disks or drums). Files are compatible because they have been created utilizing the DBMS software. Via the DBMS, application programs can access the database, retrieve the desired data from different files and process the data into meaningful information.

2. DBMS: Advantages and Disadvantages

Kroenke [Ref. 34: pp. 3-17] provides a summary of advantages and disadvantages of database systems in comparison to conventional file processing systems:

Advantages of Database Systems

1. **More information can be produced from a given amount of data.** A database consists of integrated data. With file systems, data is physically partitioned limiting the combinations of data that can be processed and hence the amount of information that can be obtained (without doing the file format conversion discussed earlier).

2. **New requests and ad hoc requests are more easily implemented.**

3. **Database systems can eliminate or minimize data duplication.** In the file processing system, some data is apt to be recorded in a number of files. With database, it need only be recorded once saving file space and to some extent, reducing processing requirements. A related problem to data duplication is data integrity. With non-integrated files, it is possible to change the data in one place but not in another. This results in data items that disagree with one another undermining the value of the information that is produced.

4. **Program/data independence can be realized.** Applications programs in the database environment access files through an intermediate DBMS which contains the descriptions of the files' data formats. If one of the data formats within a file is modified, only the DBMS and the applications programs that access the altered data files need be changed. In the file processing environment, each program contains its own set of data structures (format descriptions) that can lead to incompatibilities when a data field format is changed within any file. All programs that access a modified file must be changed regardless of whether they use the particular data item that was altered.

5. **Better data management.** Since data is centralized in a database, one department (or person) can specialize in the maintenance of data. Economies of scale can be realized by assigning one full-time person to centrally manage and control data modifications

instead of many file maintenance groups working part-time on data problems. Personnel cost savings can be spent on a more powerful and sophisticated DBMS package.

6. **Affordable sophisticated programming can be realized.** Because of the flexibility in manipulating files and user-oriented presentation languages, programming with a DBMS reduces development and maintenance costs even though the number of application programs that are written increase.

7. **Representation of record relationships.** Data items are grouped into records and a collection of records is called a file. A database system is then a collection of integrated files and the relationships among records in those files. With file processing, the absence of record relationships makes the combining of data among different files more difficult.

Disadvantages of Database Processing

1. **It can be expensive.** A DBMS product can cost more than $100,000 to buy. The package may occupy so much main memory that additional memory must be purchased. Even with adequate main memory, it may monopolize the CPU (central processing unit) forcing the user to upgrade to a more powerful computer. Conversion from file processing systems may be expensive particularly when new data is added to the data residing on existing systems. Higher operating costs may result with some database systems. Sequential processing, for example, is not done as quickly in the database environment.

2. **The database system is complex.** Increased complexity and concurrent processing make it difficult to determine the exact state of the database if a failure occurs. Backup and recovery is complicated and can be a major undertaking if the application program causing the failure has modified several records. Invalid data may be passed to other programs that read the modified records before the problem was detected and eliminated.

3. **Vulnerability to failure increases with database systems.** Centralization of data files increases vulnerability. A failure of one component of an integrated system can stop the entire system. This event can halt operations if the user group is dependent on the database.

Strictly speaking, file processing systems can achieve the same advantages that database systems have. It is possible to have a database and to apply the principles of database management without using a commercial package, but, it will require application programmers to write sophisticated and complicated data management programs. In this thesis, the acronym DBMS refers to commercially developed systems.

3. Determining a Need for a DBMS

Many organizations invest in DBMS technology because they want to provide easy access to as much data as possible, as quickly as possible. However, the database approach may not be feasible or cost-effective in all situations. There are a number of criteria that managers should consider when deciding whether their organization can benefit from a database system.

Database systems can eliminate or minimize redundant data storage required with traditional file systems. If an organization uses hundreds or thousands of data files which contain many of the same data items, the costs incurred in updating these files can be extensive. Since each file must be modified independently, data collection and verification must be carefully controlled or inconsistent data items can result among files. Inconsistencies among files can lead to differing and erroneous outputs when some systems or reports require data from two or more data files. A way to estimate the scope of this problem is to compute the number of files used by a particular system or report application. The sum of these results for all systems and report applications will indicate how many data files are involved in integrated processes. If this number is large or is expected to grow soon, the database approach would be beneficial.

The type of processing an organization does more frequently is another consideration. The production of paychecks, certain invoices and other routinized requests normally can be processed more efficiently utilizing customized programs and access methods. When the number of ad hoc inquiries begins to dominate the production of routine requests, a database system can provide a flexible and more cost-effective means to handle one-time requests. In this case, the primary advantage of database methods over traditional file systems is the ability to generate applications programs quickly in response to new requirements. An organization in which processing requirements are relatively static, e.g., one that runs mostly production systems, would gain few benefits from the database approach. [Ref. 35: pp. 10-11]

Database management systems are relatively expensive. Managers may well question the practicality of spending much money and effort to implement a DBMS if the

return on their investment is not substantiated. A typical DBMS payback curve runs negative at first due to the heavy outlay for software, planning, organizing and hiring and training new staff. Make or buy alternatives must be examined. If the organization plans to make a substantial improvement in their existing system, a commercial DBMS product is probably the best approach. Commercial systems represent larger initial costs than in-house developed systems, but, they promise long-range benefits in vendor support commensurate with state-of-the-art technological advances. In-house development of comparable software would eventually exceed the cost of a commercial system because it would require maintaining a specialized programming staff to keep up with necessary enhancements and corrections to the DBMS software. [Ref. 36: pp. 126-128]

When assessing the intangible and tangible benefits of a database system, decision makers should consider the impact on top-management, functional management and data processing management.

Top-managers should realize increased responsiveness to requests for new information. Additionally, a database management system should impact on data processing costs by reducing application development time and costs. The DBMS should be more than just a foundation for software development, it should be a foundation for running the organization. Key managers, therefore, must be convinced that their DBMS investment will provide for the comprehensive informational needs of the organization. [Ref. 36: p. 127] Functional managers should observe a trend toward decentralizatien in both development of application systems and in the use of the database. Users must be heavily involved in the initial database design to ensure that the resulting system will not be incompatible with their information needs.

An operational database should provide an effective facility to meet the user's changing requirements. Inquiry and reporting systems should be designed to allow users to manipulate the database directly. Functional user groups within the organization may need to share data. The database management system should provide the capability to integrate processes at the same time it controls access to sensitive data according to the organization's security and privacy policies. [Ref. 35: pp.11-12]

Data processing managers should consider how a DBMS will impact on operational and staffing requirements. Although database systems could reduce some hardware requirements, in terms of storage media, it is more likely that a DBMS will outgrow the present computer system's capacity. Current system's capacities and response times should be analyzed and compared to projected capacities and response times over the life cycle of the new database system. These estimates can be particularly hard to compute since it is difficult to predict the rise in user requested applications. A DBMS may also drastically increase on-line transaction processing time when they are run concurrently on the same computer system. The solution to this problem may either require shifting some of the workload to slack processing periods or by purchasing more powerful computers.

The DP manager should also estimate additional staffing requirements necessary to support the DBMS. Some organizations will not have sufficient numbers of programmers and analysts experienced with DBMS technology. Hiring or training computer professionals in this area may represent a significant long-term expense. With larger database systems, several personnel may have to be assigned the responsibility of administering database functions. Pooling individuals with technical skills may be an economic way to centrally control the database, but it may also

present an unacceptable cost to the organization's top leadership. These issues along with other technical aspects must be clearly defined so that the best system can be obtained.

4. Problems with Database Methods

While database technology offers phenomenal productivity gains and data integration benefits, it also introduces a number of problems into an organization. Many organizations have not met their development expectations using database methods because of management, end user and technical inadequacies.

The database approach requires that users supply about 40% to 50% of the total system development effort; from the planning phase through system implementation, testing, and delivery. This compares with only 10% to 20% with conventional file systems. The end user problem becomes evident when the people assigned are not available full-time or lack the proper analytical skills. Adequate funding to support user training is often overlooked or critically limited by management. [Ref. 28: p. 129]

The lack of an accurate user requirements definition also undermines the database project effort. The DP or MIS staff may be pressured by users and management to bring the system on-line before requirements are fully established. Without a complete specification to work with, the project team may resort to copying a previous database system which in turn may have been copied from its predecessor. The introduction of newer, more sophisticated database management software requires a comparable level of sophistication from end users. More often than not, end users continue to use antiquated business procedures that limit the potential gains that are achieveable with database methods. User groups may want to retain some autonomy in

the use of database resources. Untended, this situation can compromise the concept of sharing organizational data resources. [Ref. 37: pp. 187-188]

Computer professionals are apt to become entranced by DBMS technology and ignore the all-important areas of planning and standards. The use of sophisticated tools must be accompanied by rigorous standards and procedures. Standardization is a prerequisite for effective data sharing. Management and end users may forego attempts to standardize data definitions and report formats when the project schedule slips and development costs increase. Some users may skip the step of stringent DBMS package evaluation and selection. This may result in acquiring a package that is inadequate, too cumbersome, or too costly. When this happens, the users must make the difficult choice between scrapping the database project or modifying the DBMS package to suit their needs.

The consequences of any of these problems can be serious. Managers must be conscious of potential problems early in the database development effort or be willing to pay the price when things go wrong. The chief management problems that will continuely beset the database system are "people, software, people, organization and people." [Ref. 38: p. 197]

One method of organizing and standardizing data that can greatly assist a project team during the database development life cycle, is through the use of a data dictionary system (DDS). Although DDS techniques have been used for several years, their value is increasing with the expansion of database technology. The data dictionary system is primarily a development tool, but, it also has many features that provide continuing maintenance support for organizational data. The basic features of DDS and its functions are briefly described in the following section.

E. DATA DICTIONARY SYSTEMS (DDS)

Data dictionary systems can greatly enhance the management function concerning organizational data. They can be used in either database or conventional file environments. The main purpose of DDS is to support the integration of the organization's data. As such, the data dictionary system is a productivity tool that can be used by computer professionals or non-DP personnel. However, members of the DP staff, application programmers, and systems analysts will normally use the DDS more than end users.

1. Data Dictionary System Features

A DDS stores all the information about data elements, records, databases, programs, reports, transactions, organization, business functions, end user views, and other project details. It is therefore necessary that an appropriate data dictionary package be available, and that proper procedures be in place to make the dictionary useful to system developers. If an automated dictionary is not available, a manual DDS should be developed. An automated DDS consists of a database and a set of programs designed to perform some of the common processing tasks associated with the maintenance and use of metadata.[5] Traditional methods of manual documentation and cross-referencing can be used but, their use requires extensive clerical support to maintain the cross-references and to modify the metadata. [Ref. 35: pp. 179-182]

---

[5]Metadata is data about the database. It includes descriptions of the meaning of data items, the ways in which the data are used, their sources, their physical characteristics, and other rules or restrictions on their forms or uses.

2. Selecting a Data Dictionary System

Selecting a package DDS should follow a rigorous evaluation process. Data processing personnel should be the primary evaluators, since the majority of DDS features are oriented to DP technicians. If data description maintenance for a DBMS is a main objective, the DDS selected must have an interface available for this DBMS. Some DBMS vendors offer corresponding DDS products. A DBMS-oriented data dictionary may be fine for maintenance, but, it may be less capable of handling non-DBMS definitions or system development information. Data dictionary systems that are designed independent of any particular DBMS may be the best choice in an environment where a database management system has not yet been selected or where multiple DBMS packages are in use.

A final consideration in DDS selection concerns the trade-off between access control and maximum flexibility in reporting DDS database contents. The ideal mix of features is that in which application program access is provided but where this external access is monitored by the DDS to prevent unauthorized modification of the dictionary contents. [Ref. 35: p. 185]

F. SUMMARY

Sophisticated software tools are essential for effective data resource management. These packages are expensive, however, and should not be purchased or developed in-house without first conducting a thorough evaluation of their basic features. The organization's background and experience with these tools is another critical factor. Implementing, testing, and user training in the use of new software packages represents a large investment of money and time. These costs must be shared throughout the

organization since everyone can benefit from the new system. In the long run, DBMS, 4GL and query language systems will be more cost-effective than conventional file systems.

As the DON modernizes antiquated computer systems and acquires progressive development software, managers should witness a dramatic rise in productivity within their organizations. But, these advanced tools are of limited use without elementary planning and development methodologies. Navy managers who want to achieve effective use of computer resources, must incorporate the three areas of IS planning, formal development methods and advanced technology into their organizational structures.

VII. SUMMARY

The preceding chapters have outlined the basic features of information systems, presented planning and organizing concepts, and briefly surveyed current techniques for requirements analysis and specification. Additionally, key features and management issues associated with state-of-the-art applications development and database management software were discussed.

Advancements in the quality and availability of information systems resources offer Navy managers many opportunities to improve their data/information handling capabilities. Recent Congressional legislation and guidance mandate the adoption of IRM. The Executive directives clearly encourage the effective and efficient planning and control of information throughout the Federal government. OMB has decreed that Federal agencies will ease the user dependence on data processing specialists and inflexible programming languages (e.g., COBOL). These government regulations and directives, however, are inconsistent with the earlier legislation which restrains the growth of modern information systems. Strict controls over the acquisition of ADPE and other computer resources is counterproductive to the construction of advanced facilities to improve data/information management. Under present acquisition rules, long lead-times for new developments will not be responsive to Congress's urgent call to implement IRM. This suggests that the ADP acquisition life cycle should be reevaluated and modified to accommodate change.

Information systems development is a heavily labor-intensive effort. It is necessary to provide effective tools to handle those aspects of the development

process where computer power can lighten the workload. These tools, combined with traditional management methods, must be properly applied to all areas of development: planning, analysis, design, and implementation. To develop viable information systems, Navy managers must take advantage of new technologies which can improve present outputs by several orders of magnitude. Modern development methods and tools such as structured analysis, DBMS, DDS, and 4GL require intensive user involvement. However, users must have a substantial knowledge base to use the tools effectively.

Senior Navy managers must commit their organizations to the use of strategic planning, structured development methods, and productivity tools. Lower management levels must support wide-spread user education and participation in information systems developments. End user involvement is vital to the areas of IS strategic planning, specification, and design. By making end users responsible for their information system developments, some of the benefits that can result include: matching the system architecture to operational requirements; increasing user awareness of the costs and effort associated with computer projects; and minimizing low-priority or unnecessary user application requests on DP.

Automated tools, structured analysis techniques, and development methodologies are only a partial solution to the Navy's computer-oriented problems. DON managers must understand the technology and human factors that will confront them at every turn of the information system life cycle. But, as private enterprise has demonstrated, a well-designed information system can give managers the capacity and flexibility to deal with our complex and dynamic world.

APPENDIX A

QUESTIONS THAT KEY MANAGERS SHOULD ASK ABOUT DP

OVERALL EFFECTIVENESS

1. Is the DP Department working on the right problems?
   a. Who identifies the problems that are important?
   b. Who sets priorities and assigns resources?
2. Are DP users satisfied with the quality of services provided by the DP Department?
   a. How can I distinguish between legitimate user complaints and noise?
3. How do I know if our DP manager is doing an effective job?
   a. What criteria should I use to measure his effectiveness?
   b. Should I judge him as I would other functional or line managers? Or as a manager of a staff department?
4. Are we spending an appropriate amount on DP?
   a. How much do we spend relative to other organizations?
   b. Do we have any quantitative measures of return on these expenditures?
5. What role should I play in the overall direction of DP effort?
   a. What decisions should I reserve to myself?
   b. What can I delegate to users? To DP management?
6. How much do I need to know about technology to play a legitimate role in key decisions?
   a. How do I acquire this knowledge?

PLANNING

1. How can I translate organization objectives into meaningful objectives for DP?
   a. How can I involve other senior managers in this process?
2. What are the appropriate objectives for DP?
   a. Should DP be entirely service oriented?
   b. Should DP aggressively "sell" its services? Or should it respond to needs expressed by others?
3. Should we have a long-range DP plan?
   a. What should it contain?
   b. Who should review it?
   c. What time period should it cover?
   d. How often should we revise it?
4. How can I evaluate requests for expansion of our processing capabilities, facilities and/or staff?
   a. How can I balance service needs against costs?
   b. When can I expect both to level off?
5. How can I get DP to be more realistic in its planning?
   a. Have we learned from our past mistakes?
6. Do our DP plans now contain explicit assumptions about the internal and external environment?
   a. Are these assumptions ever verified? By whom?
7. Are there technological developments yet to come that will obsolete our current capabilities (including our people)?
   a. How do I plan for these and minimize their impact?
8. Are there sociological developments that will impact what we do and the cost of doing it?
   a. Do we have adequate security protection in our systems? In our facilities? In our personnel policies?
   b. Have we anticipated the requirements of likely privacy legislation?

9. Should we have a corporate DP planning committee?
   a. What should be its charter?
   b. Who should be its members?
   c. How often should it meet?

BUDGETS

1. How is our DP budget distributed?
   a. By expense category: hardware, software, personnel costs, communications?
   b. By end user: finance, personnel, administration, planning, etc?
   c. By DP function: research, development, operations, maintenance, conversion, training, internal administration?
2. How much has our DP budget increased in the past three years?
   a. What are the major components of past growth?
   b. In retrospect, were the increases worthwhile?
   c. Were they anticipated?
3. How much is the DP budget expected to increase in the next three years?
   a. What are the major components of projected growth?
   b. What concrete benefits will result?
4. Are we incurring unfavorable budget variances?
   a. What analysis of variances should I ask for?
   b. What plans do we have for bringing variances under control?
5. Should DP be a cost center or a profit center?
   a. Is our cost accounting system adequate for control of DP costs?
6. Should users pay for feasibility studies? Development? Operations? Maintenance?
   a. How should we determine the amount to be charged?
7. How should DP Department overhead be treated?
   a. Should users be charged for the cost of job re-runs? Machine failure?
   b. Should users pay for DP training? Upgrades?

8. Should users be allowed to go outside for services?
   a. Under what conditions?
   b. What role should DP management play in coordinating such efforts?
9. Should DP be allowed (or encouraged) to sell its services to outsiders?
   a. How do I avoid conflicts with internal needs?

ORGANIZATION

1. Is the organizational philosophy of the DP Department consistent with that of the overall organization?
   a. Is it consistent with our stated missions and objectives?
   b. Is it consistent with the organizational view of operating, functional, and staff managers?
2. Is the DP Department placed in the organization so that it can function effectively?
   a. Do the proper communications channels exist?
   b. Are they used?
   c. How can I improve them?
3. Do both operating units and staff departments receive adequate support?
   a. Is the DP Department viewed as captive to any particular functional area?
   b. How do I correct that perception?
4. Should DP management be invited to contribute to discussions of organizational strategy?
   a. What role should the DP manager play in these discussions?
   b. Is he qualified for this role?
5. Should we bring operating-level viewpoints to bear on short-term DP planning and priorities?
   a. Would a committee or task force approach work?
   b. If so, what should be its charter? Membership?

6. Should I be concerned about the internal organization of the DP Department?
   a. Have we reviewed it recently?
7. Are we organized to do a good job on project-type activities? On production?
   a. Can we learn anything from the way we organize other (non-DP) activities in the organization?
8. Under what conditions should DP activities be centralized? Decentralized?
   a. Are economies of scale compelling or only a rationale?
9. Have we established and adopted well-defined internal standards and procedures for project evaluation, equipment selection, documentation, programming?
   a. Are they used?

PROJECT MANAGEMENT

1. How many development projects have we undertaken in the past 3 years?
   a. How many of these were considered successful by the end users?
   b. How many were completed on time and within budget?
   c. Were any projects aborted? Why?
2. Why are development projects so difficult (and, at times, painful)?
   a. Why do they take so long?
   b. Why do they cost so much?
   c. Why is it so difficult to make simple changes?
3. How rigorous and realistic is our analysis of proposed projects?
   a. Are benefit estimates supported?
   b. Are cost estimates comprehensive?
   c. Are plans and schedules detailed and realistic?

4. Do we apply the classical techniques of investment analysis to DP projects?
   a. Which ones? Do we use them routinely?
5. Do we explicitly identify and evaluate non-technical considerations before undertaking development projects?
   a. Do we consider operational problems adequately?
   b. Do we consider the economic consequences of failure?
6. Do we explicitly consider alternative approaches to the solution of user problems?
   a. Do the alternatives inclued non-computer approaches?
7. What steps does DP management take to identify user requirements?
   a. Do users know what they want?
   b. Do they express their needs clearly?
   c. Do they change their minds too often?
8. Should users be required to cost-justify their requests?
   a. Should users be held responsible for achieving project benefits? Alone?
   b. Should DP management be held responsible for meeting cost targets? Alone?
9. What is our approach to ensuring quality and reliability?
   a. Are these considerations built in during systems design?
   b. How are they measured and controlled after systems become operational?
10. Do our long-range cost and personnel projections adequately provide for ongoing maintenance of applications programs?
   a. Have we projected their useful life?
   b. Have we projected the cost of replacing them?
11. Do our internal (and/or external) auditors have an opportunity to influence system designs?
   a. What role do they play?
   b. Do they sign off on system designs before development begins?

12. Do we routinely conduct post-implementation audits of development projects?
    a. Have such audits proven useful?
    b. What actions are taken as a result of them?

PERSONNEL MANAGEMENT

1. Do we have the proper staff for the job at hand?
    a. Do our people have the necessary skills?
    b. Are there enough of them?
2. Do we promote attractive career opportunities?
    a. Are we able to recruit outstanding individuals?
    b. Are jobs and career paths well-defined and documented?
    c. Do DP employees have opportunities for tours of duty elsewhere in the organization? Is the converse true?
    d. Is turnover a problem? What are we doing to reduce it?
3. What are we doing to avoid technological obsolescence?
    a. What measures do we have of staff competence?
    b. Do we provide challenging training opportunities?
    c. Do our personnel take advantage of them?
4. Is our compensation structure rational and fair?
5. What can I do to stimulate the DP staff's interest in the organization and its objectives?

APPENDIX B
A SYSTEMS ANALYSIS CHECKLIST

Analysis Planning

Questions

1. Are the reasons for the analysis project clearly defined in writing?
2. Are the project limits defined (e.g., resources, time, and funds)?
3. Is the completion of the system scheduled?
4. Who will perform the analysis work? Does that person have any previous experience in this application area?
5. Who are the user participants?
6. Are objectives set for the new or modified system? If so, what are they, and who set them?
7. What priority has the organization set for the project?
8. What previous systems analysis work has been performed in this application area?
9. What is the status of current systems serving the application?
10. What (if any) special legal, security, or audit considerations must be observed in this system?

Deliverables

1. A narrative definition of the project boundaries
2. A tentative work plan for the analysis work
3. A user contact list
4. A tentative resource staffing list
5. A list of existing application systems
6. A priority impact statement concerning the relative importance of the system.

User Contacts

Questions

1. Are all user participants and organizational relationships identified?
2. Do users clearly understand the current system and its operation?
3. Are legitimate user complaints about the current system documented?
   Is the impact of the complaints fully documented?
4. How much time and effort are the users willing to put into the initial analysis work?
5. Are users identified as to who are supporters of, resistant to, and indifferent to the system?
6. Do users expect any specific benefits from the resulting system?
7. Is there clearly defined top-level support for the project? If so, who constitutes this support?
   How much power do they wield?
8. Who are the key decision makers in the user environment?
9. How many user locations are there? How many people will use the system at various levels?
   What is their level of computer system experience?

Deliverables

1. An organization chart of all participating user areas, including their hierarchical relationships
2. A narrative describing the user's background and prior experience
3. Documentation of user problems with the existing system and the impact of these problems
4. A work plan of expected user participation in the analysis
5. A tentative statement of user expectations
6. A narrative on the political relationships and system support expectations of the major user participants

7. A brief history of previous data systems and procedures used in the application area
8. Identification of any other organizational systems or applications that interrelate with the proposed system

System Objectives

Questions

1. Are system objectives formally defined? Or are they loosely stated and subject to interpretation and/or later definition?
2. Will the new system have a major impact on the basic operations of the organization?
3. Will the new system replace an existing one? If so, how old is the current system? How many others preceded it?
4. Is the new system expected to cause relocation or removal of any work functions? If so, how sensitive is the issue? Who will help to combat any resistance?
5. Is an interim system required to satisfy immediate goals or to eliminate intolerable problems with the existing system?
6. Is a phased development and implementation approach feasible? Or is a one-time mass conversion required?
7. What cost can be justified? What resources can be allocated for this project?
8. How close to the state of the art is the new system expected to be?
9. How much time can users allocate for training and start-up? During what period of time?

Deliverables

1. A comprehensive statement of system objectives
2. A statement of general scope and level of project effort required, including tentative cost and resource estimates
3. A statement concerning the current system and procedures considered for change, elimination, and/or replacement
4. A general statement covering the expected project phasing and the overall team approach to the project

5. A tentative statement covering the levels and impact of anticipated organizational changes that will result from the system
6. A commentary on the roles and responsibilities of each participating user department and major user group in the desired system

Current System

Questions

1. What are the problems with the current system as evaluated by the users and the technical team?
   Do these evaluations agree?
2. How do other organizations perform similar functions? What is the current state of the art in the application area?
3. What other methods and procedures have been tried and/or used to service the application?
4. What is the detailed chronology of the current system's life?
5. What is the organization's history during the current system's life?
6. What development, maintenance, and operational costs are associated with the current system (including user efforts)?
7. Identify the name, rank, and organizational position of those who supported, built, and use the current system.
8. Identify one or more major situational failures that resulted from the current system.

Deliverables

1. A comprehensive narrative on the current system and its operation, history, and users
2. A ranked list of the current system's major faults and problems
3. A full cost analysis of the current system
4. A general statement on how the new system is related to those in other organizations or the state of the art

5. A complete collection of the documents, procedures, and other available details concerning the operation/content of the current system

Data Elements and Structures

Questions

1. Are the current data elements, files, forms, procedures, and so on thoroughly documented?
2. Are the current data elements and structures logical, consistent, and utilized?
3. How clean is the database?
4. Do users have a list of new data elements they would like to see in the new system? Is it feasible to add these data elements?
5. How much redundancy exists between the current system's database and that of other applications in the organization? Are any of the other applications a more logical repository for any elements of the database?
6. Is there enough flexibility in the current data structure to perform to meet the new system's needs?
7. How difficult will it be to convert the current database to a new one? How much error testing will be necessary to achieve a clean conversion?
8. How much maintenance is normally done on the existing database?
9. Can or should extensive data archives from this database be converted?
10. How much of the current database is actively used? By whom?
11. What significant faults or failures were encountered with the data files? How were they dealt with?
12. How many times and it what ways has the database been modified?

Deliverables

1. A comprehensive set of format and content definitions or all data elements, files, and supporting data structures

2. An evaluation of current database content, with emphasis on cleanliness, errors, unused areas redundancy, conversion, and future use
3. A list of expected changes, additions, deletions, and other modifications to data elements and structures that are anticipated for the new system
4. A summary of the major uses of the data file and its elements
5. A list of faults and failures or the existing data files

User Interviews

Questions

1. Are all users identified?
2. Is there a formal interview plan for each user level covered?
3. Are lists of questions and objectives developed for the interviews at each user level?
4. Is top management supporting and publicizing the interviews, the interview team, and the overall expectations?
   Is top management making a strong pitch for interviewee cooperation?
5. Are all interviews scheduled during acceptable time periods?
6. Are the interviewers trained in effective interview techniques?
7. Are all scheduled interviews completed? Have cancelled, interrupted, or forgotten interviews been rescheduled and conducted?
8. Have the interviewers taken adequate notes and written evaluations of each interview?
9. Have the interviewers compared notes, impressions, and other observations? Are these details documented?
10. Are interviewees given adequate feedback, such as summary reports, notes, and so on?

11. Have follow-up interviews been conducted when special problems or conditions are uncovered during the initial interviews?
12. Has management been kept informed about the interview process, any problems uncovered, and uncooperative users?

Deliverables

1. A formal interview plan
2. Documentation of interview results
3. A report summarizing the interviews that includes both consensus answers and significant variances
4. An internal analysis of user attitudes and positions vis-a-vis the system
5. A management report covering interview findings and cooperation of the participants
6. Results of test interviews along with the changes in questions, emphasis, and other interviewing guidelines
7. Explanation of any imcomplete interviews

Research on Other Systems

Questions

1. What other organizations can be surveyed regarding their approach to the subject application?
2. What (if any) proprietary packages are available that might suit the application area?
3. What (if any) trade and industry associations study or catalog the systems work of others in the same field?
4. What (if any) formal literature is available on the subject application area?
5. How much time and effort should be spent in reviewing other systems?
6. Were the reviews of other systems productive? Should more time be spent on this activity?
7. Are field interviewers of other users and organizations necessary?

Deliverables

1. A list of organizations and sources to review for base knowledge on alternative approaches to the application
2. A narrative report detailing the ways other organizations are solving the application
3. A technical evaluation covering the current state-of-the-art application area
4. A summary report on contacts to other users and organizations
5. A follow-up plan for reviewing or tracking major developments in the industry

Alternative Propositions

Questions

1. How many application alternatives should be considered? How much time and effort should be spent in evaluation of alternatives?
3. How detailed and complete should the considerations of each alternative be?
4. How will the alternatives be developed and documented?
5. Are formal requirements and evaluation criteria established for the alternatives?
6. Who will evaluate the alternatives? Will the users review the alternatives?
7. Are all logical alternatives being considered?
8. Are outside expert opinions being sought on the alternatives?
9. Are the alternatives considered consistent with those evaluated by other organizations?

Deliverables

1. Alternative design definitions
2. Positive and negative factors of each alternative
3. Evaluation reports from each group that studies the alternatives
4. Formal user presentation of the alternatives
5. Preliminary cost predictions for each alternative

6. A technology impact assessment for each alternative
7. A user impact assessment for each alternative

Selecting a Design Alternative

Questions

1. Are all alternatives fully reviewed and evaluated?
2. Are the alternatives ranked in terms of their ability to meet the system requirements criteria?
3. Is there a technical-management team with authority to select the most appropriate alternative?
4. Does one alternative clearly outrank the others?
5. Which alternatives(s) do the users support?
6. Which alternative is best to implement in terms of time, cost, resources, and technical risk?
7. Which alternative uses the most advanced concepts?
8. Which alternative is likely to last the longest?

Deliverables

1. A detailed comparison of alternatives
2. A ranking of alternatives
3. A specific recommendation as to the alternative that is best to pursue
4. A report to the users on the alternative selected
5. A summary of reasons for rejecting other alternatives

Structural Analysis

Questions

1. Are all data elements, flows, and expected processing steps defined for the selected alternative?
2. Are procedural and organizational changes that the new system will generate defined and evaluated?
3. Are the content and uses of input files and outputs defined in a general way?
4. Are the equipment requirements for the new system estimated?
5. Is there a list of expected system modules?
6. Is there a tentative data conversion plan?

7. Is there an overall system flow being generated?
8. Are associated clerical procedures outlined?
9. What is the estimated volume of data and transactions?
10. Are the security and accuracy requirements of the data being considered?
11. Are testing procedures for the new approach thoroughly defined?
12. Is a preliminary system implementation plan available?

Deliverables

1. A report of the proposed system approach
2. A system flowchart
3. A user operations and responsibility flowchart
4. A detailed report on the analysis findings
5. A cost-benefit analysis report
6. A preliminary testing plan
7. A tentative implementation plan

Plans for the Next Phase

Questions

1. Are there work tasks and resource estimates for the general design work?
2. Is there a resource loading plan that shows requirements by work task?
3. Are user support tasks identified and planned? Are the users aware of them?
4. Are target dates set to obtain authorization to proceed with the next phase?
5. What is the expected completion date of the proposed work?

Deliverables

1. The work plan and the resource estimates
2. The user support plan
3. A narrative on the approach to managing the next phase

Management Presentations and Reviews

Questions

1. Are all levels of management in the technical and user areas briefed on the analysis results and recommendations?
2. Are the presentations clearly and logically formulated?
3. Are management's concerns and questions documented and answered?
4. Has the proposed alternative survived management's scrutiny?
5. Does the analysis team have any doubts about the project approach?
6. Have minority opinions and negative comments been properly addressed?

Deliverables

1. Presentation critiques and internal reviews
2. Presentation reports and visual aids
3. Authorization to proceed

APPENDIX C

REPRESENTATIVE SUPPLIERS OF DBMS AND 4TH GENERATION LANGUAGES

PRODUCT/VENDOR	ENVIRONMENT	TYPE
ADF & DMS IBM 1133 Westchester Ave. White Plains, NY 10604	IBM 370/4300/30XX DOS/VSE, MVS, CICS IMS/DC	4GL
ADS/O Cullinet Software 400 Blue Hill Dr. Westwood, MA 02090	IBM 370/4300/30XX DOS, VM/CMS, MVS	Query, 4GL, uses IDMS DBMS
APS Sage systems Inc. 3200 Monroe St. Rockville, MD. 20852	IBM 370/4300/30XX	A 4GL system which uses Database Painter DBMS, Screen Painter 4GL
ASI Inquiry Applications Software 21515 Hawthorne Blvd. Torrance, CA 90503	IBM 370/4300/30XX CICS/TSO/CMS	Query
dBase II Ashton-Tate 10150 W. Jefferson Culver City, CA 90280	CP/M-80, -86 MS-DOS, PC-DOS	DBMS, 4GL
CA-Universe Computer Associates 125 Jericho Tnpke. Jericho, NY 11753	IBM 370/4300/30XX DOS/VSE, MVS, VM/CMS, IBM PC, PC-DOS	DBMS, Query, w/Apps Form Driver 4GL

PRODUCT/VENDOR	ENVIRONMENT	TYPE
Condor Series Condor Computer Corp. 2051 S. State St. Ann Arbor, MI 48104	8080, 8085, Z80, 8086 CP/M, MP/M, MS-DOS	DBMS, 4GL Screeneditor
Conquer Sydney Development Corp. 600-1385 W. 8th Ave. Vancouver, BC V6H3V9	IBM 370/4300/30XX MVS/TSO, VM/CMS	DBMS, Query, 4GL
Data Base + Tominy 4221 Malsbary Rd. Cincinnati, OH 45242	IBM 370/4300/43XX 34, 36, IBM-PC,	DBMS, Query, 4GL
Day One Day One Software 618 Shoemaker Rd. King of Prussia, PA 19406	IBM-PC, Apple II, TRS80, Kaypro, Televideo, Compaq	DBMS, 4GL
Dayflo Dayflo Software, Inc. 2500 Michigan Dr. Irvine, CA 92715	IBM-PC & XT	DBMS, Query, 4GL
DNA-4 Exact Systems & Programming 1 Labriola Court Armonk, NY 10504	Data General MV & Nova-Eclipse RDOS, AOS, AOS M68000	DBMS, Query, 4GL

PRODUCT/VENDOR	ENVIRONMENT	TYPE
Express Mgt. Decisions Systems 200 Fifth Ave. Waltham, MA. 02254	IBM 370/4300/30XX Express E300	DBMS, Query
Falcon Peregrine Systems 15530 Rockfield Blvd. Irvine, CA 92714	IBM 370/4300/30XX DEC VAX, Unix, IBM-PC	DBMS, Query
Focus Information Builders 1250 Broadway New York, NY 10001	IBM 370/4300/30XX DOS, VM, MVS, Wang, IBM-PC, TI, PC-DOS MS-DOS	DBMS, Query, 4GL
Ideal Applied Data Research Route 206 & Orchard Rd. Princeton, NJ 08540	IBM 370/4300/30XX DOS, MVS, VM/CMS	4GL
Imagine, Accolade Multiplications, Inc. 1050 Mass. Ave. Cambridge, MA 02138	IBM 370/4300/30XX MVS, DOS/VSE, VSI	Imagine, a DBMS uses the Accolade 4GL
Info Henco Software Inc. 100 Fifth Ave. Waltham, MA 02154	IBM 370/4300/30XX VM/CMS, Prime, DEC VAX, Harris, Honeywell DPS6	DBMS, Query, uses Info 4GL

PRODUCT/VENDOR	ENVIRONMENT	TYPE
Infocen 3CI 155 W. Harvard Fort Collins, CO 80525	DEC VAX, DG MV	DBMS, Query, 4GL
Informix 3.0, Ace, Perform Relational Database Systems 2471 E. Bayshore Rd. Ste. 600 Palo Alto, CA 94303	DEC VAX, PDP-11, IBM, Altos, MS-DOS Lisa Unix, PC-DOS	Informix DBMS Informer query Perform 4GL
Ingres Relational Technology 2855 Telegraph Ave. Berkeley, CA 94705	DEC VAX, VAX/VMS, Unix, PDP-11	DBMS, uses QueL Query, ABF 4GL
Inquire Infodata Systems 5205 Leesburg Pk. Falls Church, VA 22041	IBM 370/4300/30XX DOS/VSE, MVS, OS/VSI	DBMS, 4GL
Intellect Artificial Intelligence Corp. 100 Fifth Ave. Waltham, MA 02254	IBM 370/4300/30XX DOS/VSE, MVS, VM/CMS	Query
IP-3 Computing Productivity Rte. 1-433-A Waitsfield, VT 05673	IBM 370/4300/30XX MVS	4GL which generates IMS/VS DBMS

PRODUCT/VENDOR	ENVIRONMENT	TYPE
Knowledgeman Micro Data Base Systems P.O. Box 248 Lafayette, IN 47902	IBM-PC, Victor, Altos, PC-DOS, CP/M-86, MP/M-86	DBMS, Query, Kpaint 4GL
Linc Burroughs Co. One Burroughs Pl. Detroit MI 48232	B1700-B1900 B2700-B4900 B6700-B7900	4GL
Logix Softshell Logical Software Inc. 55 Wheeler St. Cambridge, MA 02138	Z8000, 8086, 68000, PD11, Unix, Venix, Xenix	DBMS, uses Quick Q, Compl Q, Query, Select Q, 4GL editor
MAG/Base 1,2,3 MAG Software, Inc. 21054 Sherman Way Ste. 305 Canoga Park, CA 91303	DECmate II, Rainbow & CP/M, CP/M-86, IBM PC/XT Unix, PC-DOS, MP/M	DBMS Base 2,3 have 4GL
Magnum Tymshare 20705 Valley Green Cupertino, CA 95014	DEC 10, 20, Tops-20, DEC VAX, VMS	DBMS, Query, 4GL
Mantis Cincom Systems 2300 Montana Ave. Cincinnati, OH 45211	IBM 370/4300/30XX DOS, MVS, VM/CMS	4GL, uses Total, VSAM DBMS

PRODUCT/VENDOR	ENVIRONMENT	TYPE
Mapper Sperry Univac Box 500 Blue Bell, PA 19424	Entire 1100 line	DBMS, Query, 4GL
Mark V Informatics General 21050 Vanowen St. Canoga Park, CA 91304	IBM 370/4300/30XX	4GL, uses Inquiry IV Query
Metafile Sensor Based Systems 15 E. Second St. Chatfield, MN 55923	IBM-PC & compatibles PC-DOS	DBMS, 4GL
Model 204 Computer Corp. of America 675 Mass. Ave., 8th Floor Cambridge, MA 02139	IBM 370/4300/30XX MVS	DBMS, Query, User Language 4GL
Natural Software AG of N.A. 11800 Sunrise Valley Reston, VA 22091	IBM 370/4300/30XX DOS, MVS, VM/CMS	Query, 4GL uses Adabas DBMS
Nomad 2 D & B Computing Services 187 Danbury Rd. Wilton, CT 06897	IBM 370/4300/30XX VM/CMS	DBMS, Query, 4GL

PRODUCT/VENDOR	ENVIRONMENT	TYPE
NPL Info. Mgt. System Desktop Software Corp. 228 Alexander St. Princeton, NJ 08540	IBM PC & XT, Vic 9000, Apple, Burroughs B20, Sage II, DEC 350 & Rainbow, HP 9816	DBMS, 4GL
Oracle Oracle Corp. 2710 Sand Hill Rd. Menlo Park, CA 94025	IBM 370/4300/30XX DG MV, DEC VAX, PDP-11, Harris, Stratus, 68000, 8086	DBMS, 4GL, uses SQL query
Pacbase CGI Systems, Inc. 8200 Greensboro Dr. Ste. 1010 McLean, VA 22102	IBM 370/4300/30XX DOS/VS/VSE, CICS, VSAM, DL/1, IMS DB/DC, OS/VS/MVS	4GL, uses IMS/DB Codasyl DBMS
Pearlsoft Pearlsoft Division 3700 River Rd. N. Ste. 3 Salem, OR 97303	Z80a, 8080, 8085, CP/M 2.2	DBMS, 4GL
Powerhouse Cognos Systems Ltd. 275 Slater St. Ottawa, Ontario Canada K1P5H9	HP3000, DEC VAX, DG MV	A 4GL system which uses Quiz Query, Quick 4GL
Pro-IV Pro-IV 119 Russell St. Littleton, MA 01460	DEC VAX, PDP-11, RSX11M, RSTS/E, 8088/8086, 68000	DBMS, 4GL

PRODUCT/VENDOR	ENVIRONMENT	TYPE
Queo-IV Computer Techniques 1622 Main Ave. Olyphant, PA 18447	Prime 250-9950 Primeos	DBMS, 4GL
Ramis II Mathematica P.O. Box 2392 Princeton, NJ 08540	IBM 370/4300/30XX DOS, VM/CMS, MVS	DBMS, uses English Query, uses Ramis 4GL
Rapport Logica Inc. 666 Third Ave. New York, NY 10017	IBM, Burroughs, Most minis, Z80a, 8086 micros	DBMS, uses Rasql uses Rasql Query, uses IQI 4GL
Revelation Cosmos P.O. Box AH Morton, WA 98356	any Pick, PC & XT, Eagle 1600	DBMS, 4GL
Rexcom Rexcom Corp. 9575 Katy Freeway, Ste. 320 Houston, TX 77024	IBM 370/4300/30XX CDC, Prime, SEL, Harris, VAX	DBMS, uses Select Query
Rim, R:Base 4000 & 8000 MicroRim Inc. 1750 112th St., NE Bellevue, WA 98004	PC-DOS, CTOS, Unix, BTOS	DBMS, Query

PRODUCT/VENDOR	ENVIRONMENT	TYPE
Salvo Software Automation 14333 Proton Rd. Dallas, TX 75238	Several 8- & 16-bit CP/M, CP/M-86, MS-DOS	DBMS, 4GL
Speed II The Office Manager 127 SW 156th St. P.O. Box 66596 Seattle, WA 98166	Wang VS, Wang 2200	Query, 4GL
System W Comshare 3001 S. State St. Ann Arbor, MI 48106	IBM 370/4300/30XX MVS/TSO, VM/CMS, IBM-PC, PC-DOS, CP/M-86	A Decision Support System with 4GL
Themis Frey Associates Chestnut Hill Rd. Amherst, NH 03031	DEC VAX, VMS	Query
Umbrella Hogan Systems, Inc. 5080 Spectrum Dr. Dallas, TX 75248	IBM 370/4300/30XX DOS/VSE, MVS, OS/VSI	4GL

LIST OF REFERENCES

1. Amitrani, Eugene J., "Systems Masters or System's Slaves?", Forum, March 1983.

2. Hannan, James, Ed., A Practical Guide to Data Processing Management, Van Nostrand-Reinhold Company, 1982.

3. Brown, Gary D. and Sefton, Donald H., "The Micro vs. The Applications Logjam," Datamation, v. 30, n.1, January 1984.

4. Powers, Michael J., Adams, David R. and Mills, Harlan D., Computer Information Systems Development: Analysis and Design, South-Western Publishing Co., 1984.

5. Sprague, Ralph H., Jr. and Carlson, Eric D., Building Effective Decision Support Systems, Prentice-Hall, Inc., 1982.

6. Kull, David, "Don't Fall Behind with an Obsolete System", Computer Decisions, v. 16, n. 2, February 1984.

7. Head, R., Federal Information Systems Management: Issues and New Directions, The Brookings Institute, Washington, D.C., 1982.

8. U.S. Government Printing Office, "Establishing the Federal Information Resource Management Regulation," Federal Register, v. 49, n. 97, 17 May 1984.

9. Whieldon, David, "Salary-status Survey, Part I: Where the Dollars Are," Computer Decisions, v. 16, n. 2, February 1984.

10. Nolan, Richard L., "Controlling the Costs of Data Services," Harvard Business Review, July-August 1977.

11. Tichy, Noel M., Managing Strategic Change, John-Wiley and Sons, Inc., 1983

12. Beckhard, Richard and Harris, Reuben T., Organizational Transitions: Managing Complex Change, Addison-Wesley Publishing Co., 1977.

13. DeMarco, T., Structured Analysis and System Specification, Yourdon Press, 1978.

14. Dickover, M.E., McGowan, C.L. and Ross, D.T., "Software Design Using SADT," Proceedings of the ACM National Conference, 1977.

15. Ross, D.T., "Structured Analysis: A Language for Communicating Ideas," IEEE Transactions Software Engineering, January 1977.

16. Teichroew, D., and Hershey, E., "PSL/PSA: A Computer Aided Technique for Structured Documentation and Analysis of Information Processing Systems," IEEE Transactions Software Engineering, January 1977.

17. International Business Machines Corporation, Business Systems Planning Guide, 4th Ed., 1984.

18. Fort Ord Automation Management Office report, Information Systems Planning Study, December 1983.

19. Shaw, Dean L., "A Total Systems Approach," Infosystems, v. 31, n. 11, November 1984.

20. Wetherbe, James C., Systems Analysis and Design: Traditional, Structured, and Advanced Concepts and Techniques, 2nd Ed., West Publishing Co., 1984.

21. Wenig, R., "Systems Analysis Checklist," System Development Management, Auerbach Publishers, Inc., 1981.

22. Gane, Chris and Sarson, Trish, Structured Systems Analysis: Tools and Techniques, Prentice-Hall, Inc., 1979.

23. Pressman, Roger S., Software Engineering: A Practitioner's Approach, McGraw-Hill Book Co., 1982.

24. Cash, James I., Jr., McFarlan, F. Warren and McKenney, James L., Corporate Information Systems Management: Text and Cases, Richard D. Irwin, Inc., 1983.

25. Smallowitz, Howard, "Avoiding Pitfalls and Costly Detours," Computer Decisions, v. 16, n. 16, December 1984.

26. DeMarco, Tom, Controlling Software Projects, Yourdon Press, 1982.

27. Blanpied, Greg, "Using Fourth Generation Languages Well," Software News, v. 4, n. 7, July 1984.

28. Snyders, Jan, "In Search of a 4th Generation Language," Infosystems, v. 31, n. 10, October 1984.

29. Santarelli, Mary-Beth, "What Kind of Aid? It Depends on Skill.", Software News, v. 4, n. 7, July 1984.

30. Cantral, David, "Speaking the User's Language," Software News, v. 4, n. 7, July 1984.

31. Betts, Ritch, "OMB: Improve DP Investments, Cut Software Costs," Computerworld, February 25, 1985.

32. Sweet, Frank, "What, If Anything is a Relational Database?", Datamation, v. 30, n. 11, July 1984.

33. Martin, James, Managing the Database Environment, Prentice-Hall, Inc., 1983.

34. Kroenke, David, Database Processing: Fundamentals, Design, Implementation, 2nd ed., Science Research Associates, Inc., 1983.

35. Weldon, Jay-Louise, Database Administration, Plenum Press, 1981.

36. Snyders, Jan, "DBMS: The More You Get, The More You Want", Computer Decisions, v. 16, n. 2, February 1984.

37. Shah, Arvind D., "Data Administration: It's Crucial", Datamation, v. 30, n. 1, January 1984.

38. Sweet, Frank, "Wanted: Experienced Kamikaze Pilots", Datamation, v. 30, n. 1, January 1984.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station, Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5100	2
3.	Department Chairman, Code 54 Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943-5100	1
4.	CDR Frank E. Royer, Code, Code 54Ro Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943-5100	1
5.	Dr. Norman Lyons, Code 54DH Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943-5100	1
6.	CAPT C. E. Baker, Code 05 Naval Finance Center 1240 E. 9th Street Cleveland, Ohio 44199	1
7.	Mr. Don Collins, Code 51 Naval Finance Center 1240 E. 9th Street Cleveland, Ohio 44199	1
8.	LT Gregory S. Curtis 104 Masonic St. Rockland, Maine 04841	2
9.	Computer Technology Programs, Code 37 Naval Postgraduate School Monterey, California 93943-5100	1

214153

Thesis
C9581
c.1

Curtis
Managing computer systems development: understanding the human and technological imperatives.

thesC9581
Managing computer systems development:
3 2768 000 62793 9
DUDLEY KNOX LIBRARY